Time Dependent Discrete Ordinates Neutron Transport Using Distribution Iteration in XYZ Geometry

DISSERTATION

James R. Dishaw, Major, USAF

AFIT/DS/ENP/07-S01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/DS/ENP/07-S01

Time Dependent Discrete Ordinates Neutron Transport Using Distribution Iteration in XYZ Geometry

DISSERTATION

Presented to the Faculty
Department of Engineering Physics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

James R. Dishaw, B.A., M.S.
Major, USAF

September 2007

AFIT/DS/ENP/07-S01

Time Dependent Discrete Ordinates Neutron Transport Using Distribution Iteration in XYZ Geometry

James R. Dishaw, B.A., M.S.
Major, USAF

Approved:

/signed/ 3 October 2007

K. A. Mathews (Research Advisor) Date
Professor of Nuclear Engineering

/signed/ 3 October 2007

R. O. Baldwin (Dean's Representative) Date
Associate Professor of Computer Engineering

/signed/ 3 October 2007

W. P. Baker (Member) Date
Associate Professor of Mathematics

/signed/ 3 October 2007

D. W. Gerts (Member) Date
Adjunct Assistant Professor of Nuclear Engineering

Accepted:

/signed/ 12 October 2007

M. U. Thomas Date
Dean, Graduate School of
Engineering and Management

AFIT/DS/ENP/07-S01

Abstract

The distribution iteration (DI) algorithm, developed by Wager [32] and Prins [28], for solving the Boltzmann Transport Equation (BTE) has proven, with further development, to be a robust alternative to von Neumann iteration on the scattering source, aka source iteration (SI). Previous work with DI was based on the time-independent form of the transport equation. In this research, the DI algorithm was

1. Improved to provide faster, more efficient, robust convergence
2. Extended to XYZ geometry
3. Extended to Multigroup Energy treatment
4. Extended to solve the time-dependent form of the Boltzmann Transport Equation.

The discrete ordinates equations for approximating the BTE have been solved using SI since the discrete ordinates method was developed at Los Alamos Scientific Laboratory by 1953. However, SI is often inefficient by itself and requires an accelerator in order to produce results efficiently and reliably. The acceleration schemes that are in use in production codes are Diffusion Synthetic Acceleration (DSA) and Transport Synthetic Acceleration (TSA). DSA is ineffective for some problems, and cannot be extended to high-performance spatial quadratures. TSA is less effective than DSA and fails for some problems. Krylov acceleration has been explored in recent years, but has many parameters that require problem-dependent tuning for efficiency and effectiveness.

The DI algorithm is an alternative to source iteration that, in our testing, does not require an accelerator. I developed a formal verification plan and executed it to verify the results produced by my code that implemented DI with the above features. A new, matrix albedo, boundary condition treatment was developed and implemented so that infinite-medium benchmarks could be included in the verification test suite. The DI algorithm was modified for parallel efficiency and the prior instability of the refinement sweep was corrected. The testing revealed that DI performed as well or faster than source iteration with DSA and that DI continued to work where DSA failed. Performance did degrade when the diamond-difference (without fixup) spatial quadrature was used. Because diamond-difference is a non-positive spatial quadrature, it can produce nonphysical negative fluxes, particularly in higher dimensions. I developed a new fixup scheme to accommodate the negative fluxes, but it did not improve performance in XYZ geometry when the scattering ratio was near unity.

My DI algorithm successfully solves the time-dependent form of the BTE using the semi-implicit method implemented by PARTISN. The agreement between DI and PARTISN was excellent.

With these improvements and tests, DI is ready for use as a general replacement for von Neumann iteration on the scattering source.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

Table Page

List of Algorithms

Algorithm Page

List of Symbols

Symbol		Page
$\vec{r}$	Position vector	1
$\hat{\Omega}$	Direction of travel	1
E	Energy	1
t	Time	1
v	Neutron velocity	1
σ	Total cross section	1
ψ	Angular flux	1
q	Rate density	1
σ_s	Differential scattering cross section	1
χ	Fission spectrum	1
ν	Mean number of neutrons produced per fission	1
σ_f	Fission cross section	1
q^{e}	Fixed emission source	1
q^{s}	Scatter source	1
q^{f}	Fission source	1
P_l	Legendre polynomial	3
w_N	Angular Quadrature weight	4
ψ_g	Angular flux at energy group g	4
g	Energy group index	4
v_g	Group velocity	5
c	Scattering-to-total cross section ratio	7
$\vec{J}$	Vector form of the partial currents	20
$\vec{J}^{\mathrm{e}}$	Vector form of the emission source	20
$\mathbb{M}$	Transit probability matrix	20
$\mathbb{I}$	Identity matrix	21
$\mathbb{G}$	Global transport matrix	21

Symbol		Page
ψ^{A}	Cell average angular flux	21
$\vec{j}^{\mathrm{in}}$	Cell inflow currents	21
$\vec{j}^{\mathrm{out}}$	Cell outflow currents	21
$\mathbb{K}^{\mathrm{OI}}$	Cell inflow to cell outflow matrix	21
$\mathbb{K}^{\mathrm{OE}}$	Cell emission source to cell outflow matrix	21
$\mathbb{K}^{\mathrm{OS}}$	Cell scattering source to cell outflow matrix	22
$\mathbb{K}^{\psi\mathrm{I}}$	Cell inflow to cell average flux matrix	22
$\mathbb{K}^{\psi\mathrm{E}}$	Cell emission source to cell average flux matrix	22
$\mathbb{K}^{\psi\mathrm{S}}$	Cell scattering source to cell average flux matrix	22
Σ	Cell scattering matrix	22
ζ	Inward flux distribution	23
$\mathbb{Z}$	Inward flux distribution matrix	23
J_i^+	Partial current, $\mu > 0$	25
J_i^-	Partial current, $\mu < 0$	25
ψ^{j+1}	Angular flux at t_{j+1}	32
ψ^j	Angular flux at t_j	32
Δt	Time interval	32
$\tilde{\sigma}_{\mathrm{t}_g}$	Synthetic total cross section	33
$\tilde{q}_g$	Synthetic source	33
$\bar{q}$	Cell average source	38
$\mathbb{A}$	Matrix Albedo	48
$\vec{j}_L$	Left face current	53
$\vec{j}_R$	Right face current	53
$\vec{j}_B$	Bottom face current	53
$\vec{j}_T$	Top face current	53
$\vec{j}_P$	Back (posterior) face current	53
$\vec{j}_F$	Front face current	53
$\bar{q}^{\mathrm{s}}$	Cell average scattering source	53

Symbol		Page
$\bar{q}^{\mathrm{e}}$	Cell average emission source	53
Σ_{s}	Isotropic scattering matrix	55
$\Gamma(a, z)$	Upper incomplete gamma function	120
$H(z)$	Heaviside step function	122

List of Abbreviations

Abbreviation		Page
BTE	Boltzmann Transport Equation	1
SI	Source Iteration	7
DI	Distribution Iteration	8
DD	Diamond Difference	17
SC	Step Characteristic	17
LD	Linear Discontinuous	19
LC	Linear Characteristic	19
V&V	Validation & Verification	61
MFP	Mean Free Path	64
DSA	Diffusion Synthetic Acceleration	75
TSA	Transport Synthetic Acceleration	75
ISS	Isotropic Surface Source	121

Time Dependent Discrete Ordinates Neutron Transport Using Distribution Iteration in XYZ Geometry

I. Introduction

The linearized form of the Boltzmann Transport Equation (BTE), aka the neutron transport equation, can be expressed as

$$\left[\frac{1}{v}\frac{\partial}{\partial t} + \hat{\Omega}\cdot\vec{\nabla} + \sigma(\vec{r},E,t)\right]\psi(\vec{r},\hat{\Omega},E,t) = q(\vec{r},\hat{\Omega},E,t), \tag{1.1}$$

where $\vec{r}$ is the position; $\hat{\Omega}$ is the direction of travel; E is the energy of the neutron; t is time; v is the neutron velocity; σ is the total cross section and ψ is the angular flux. The rate density, q, can be represented as

$$\begin{aligned} q(\vec{r},\hat{\Omega},E,t) = &\int_0^\infty \mathrm{d}E' \int \mathrm{d}\Omega'\ \sigma_s(\vec{r},E' \to E,\hat{\Omega}'\cdot\hat{\Omega},t)\psi(\vec{r},\hat{\Omega}',E',t) \\ &+ \chi(E)\int \mathrm{d}E'\ \nu\sigma_f(\vec{r},E',t)\int \mathrm{d}\Omega'\ \psi(\vec{r},\hat{\Omega}',E',t) + q^{\mathrm{e}}(\vec{r},\hat{\Omega},E,t), \end{aligned} \tag{1.2}$$

where σ_s is the differential scattering cross section; χ is the fission spectrum; ν is the mean number of neutrons produced per fission; σ_f is the fission cross section (Lewis & Miller [22]) and q^{e} is the fixed (embedded) emission rate density. The rate density can be further categorized as

$$q^s(\vec{r},\hat{\Omega},E,t) = \int_0^\infty \mathrm{d}E' \int \mathrm{d}\Omega'\ \sigma_s(\vec{r},E' \to E,\hat{\Omega}'\cdot\hat{\Omega},t)\psi(\vec{r},\hat{\Omega}',E',t) \tag{1.3}$$

$$q^f(\vec{r},\hat{\Omega},E,t) = \chi(E)\int \mathrm{d}E'\ \nu\sigma_f(\vec{r},E',t)\int \mathrm{d}\Omega'\ \psi(\vec{r},\hat{\Omega}',E',t) \tag{1.4}$$

where q^s is the scattering source and q^f is the fission source.

The vast majority of all research in neutron transport is focused either on the time-independent BTE or on Monte Carlo methods. The time-independent BTE simplifies the transport equation by solving for the steady-state solution, which forces the time derivative to zero. A common approach for solving the time-dependent BTE utilizes a method based on solving the time-independent BTE (Dupree [11], Reed [30] and Hill [16]). Monte Carlo methods are also used to solve time-dependent problems; however, they can require a large number of particle histories in order to reduce uncertainty.

The focus of this research was to incorporate the distribution iteration method into a multigroup, time-dependent BTE solver. The distribution iteration algorithm has substantial advantages over the more common source iteration algorithm, the most notable ones being its rapid convergence in highly-scattering regions and the ability to easily implement distribution iteration in parallel.

1.1 Background

The integro-differential form of the BTE makes it particularly difficult to solve and analytic solutions are only possible for the simplest of problems. Numerical solutions to the BTE are obtained by applying a variety of approximations in order to simplify the BTE (Carlson [7]):

- Assume a stationary (time-independent) solution
- Discretize in energy (multigroup approximation)
- Discretize in angle (discrete ordinates)
- Spherical harmonics expansion of the scattering source

The direct application of difference methods directly to the BTE is usually avoided because neutrons are not strictly conserved (Carlson [9]). In order to ensure neutron conservation, cell averages of the distribution function are used and cell balance equations are coupled to the system of equations.

The spherical harmonics expansion of the scattering source is the result of using a Legendre expansion of the differential cross section σ_s in the scattering source q^s

making the expansion

$$\sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega}, t) = \sum_{l=0}^{\infty}(2l+1)\sigma_{sl}(\vec{r}, E' \rightarrow E, t)P_l(\hat{\Omega}' \cdot \hat{\Omega}), \tag{1.5}$$

where P_l is the Legendre polynomial of order l and σ_{sl} are the Legendre scattering moments. Substituting this approximation into the scattering source yields,

$$q^s(\vec{r}, \hat{\Omega}, E, t) = \sum_{l=0}^{L}(2l+1)\int \mathrm{d}E'\ \sigma_{sl}(\vec{r}, E' \rightarrow E, t) \int \mathrm{d}\Omega'\ P_l(\hat{\Omega} \cdot \hat{\Omega}')\psi(\vec{r}, \Omega', E', t). \tag{1.6}$$

where the Legendre expansion has been truncated to L terms. Using the Legendre addition theorem,

$$P_l(\hat{\Omega} \cdot \hat{\Omega}') = \frac{1}{2l+1}\sum_{m=-l}^{l} Y^*_{lm}(\hat{\Omega})Y_{lm}(\hat{\Omega}'), \tag{1.7}$$

where Y_{lm} is the spherical harmonics function and Y^*_{lm} is its complex conjugate, permits the removal of the inconvenient scattering angle expression, $\hat{\Omega}' \cdot \hat{\Omega}$, and replaces it with the more convenient directions of travel $\hat{\Omega}'$ and $\hat{\Omega}$. With the above derivations, the scattering term can be expressed as

$$q^s(\vec{r}, \hat{\Omega}, E, t) = \sum_{l=0}^{L}\sum_{m=-l}^{l} Y^*_{lm}(\hat{\Omega})\int \mathrm{d}E'\ \sigma_{sl}(\vec{r}, E' \rightarrow E, t) \int \mathrm{d}\Omega'\ Y_{lm}(\hat{\Omega}')\psi(\vec{r}, \hat{\Omega}', E', t). \tag{1.8}$$

The angular flux coefficients are defined as

$$\phi_{lm}(\vec{r}, E', t) = \int \mathrm{d}\Omega'\ Y_{lm}(\Omega')\psi(\vec{r}, \hat{\Omega}', E', t), \tag{1.9}$$

which yields

$$q^s(\vec{r},\hat{\Omega},E,t) = \sum_{l=0}^{L}\sum_{m=-l}^{l} Y_{lm}^*(\hat{\Omega}) \int \mathrm{d}E' \; \sigma_{sl}(\vec{r},E' \to E,t)\phi_{lm}(\vec{r},E',t). \tag{1.10}$$

This form of the scattering term when combined with an angular quadrature, the discrete ordinates approximation, results in a finite set of equations where the angular dependence has been explicitly eliminated. Given an angular quadrature of N ordinates, the angular flux coefficients can be determined from

$$\phi_{lm}(\vec{r},E',t) = \begin{bmatrix} w_1 Y_{lm}(\hat{\Omega}_1) \ldots w_N Y_{lm}(\hat{\Omega}_N) \end{bmatrix} \begin{bmatrix} \psi_1(\vec{r},E',t) \\ \vdots \\ \psi_N(\vec{r},E',t) \end{bmatrix}, \tag{1.11}$$

where the matrix on the left has rows of l and m and w_N are the angular quadrature weights. In slab geometry, the spherical harmonics functions reduce to Legendre polynomials due to symmetry. In XY geometry, the spherical harmonics functions reduce to just the even terms. However, in XYZ geometry the spherical harmonic functions do not simplify.

Having discretized the angular dependence, we now seek to discretize the energy dependence. The energy discretization approach used in almost all deterministic methods is the multigroup approximation, which divides the continuous energy range into a finite set of groups. This transforms the integration over energy into a summation over the energy groups. The multigroup approximation defines the group angular flux (ψ_g) as

$$\psi_g(\vec{r},\hat{\Omega},t) = \int_{E_g}^{E_{g-1}} \mathrm{d}E \; \psi(\vec{r},\hat{\Omega},E,t) = \int_g \mathrm{d}E \; \psi(\vec{r},\hat{\Omega},E,t), \tag{1.12}$$

where g is the energy group index such that $g \in [1,G]$, where G is the total number of energy groups. The above definition transforms the BTE into a coupled set of monoenergetic transport equations, which are then solved by iteration.

The first step in deriving the within-group transport equation is the integration of equation (1.1) over an energy group, g, yielding

$$\begin{aligned}\left[\frac{1}{v_g}\frac{\partial}{\partial t} + \hat{\Omega}\cdot\vec{\nabla}\right]\int_g \mathrm{d}E\ \psi(\vec{r},\hat{\Omega},E,t) + \int_g \mathrm{d}E\ \sigma(\vec{r},E,t)\psi(\vec{r},\hat{\Omega},E,t) \\ = \sum_{g'=1}^{G}\int_g \mathrm{d}E\ \int_{g'}\mathrm{d}E'\ \int \mathrm{d}\Omega'\ \sigma_s(\vec{r},E'\to E,\hat{\Omega}'\cdot\hat{\Omega},t)\psi(\vec{r},\hat{\Omega}',E',t) \\ + \int_g \mathrm{d}E\ \chi(E)\sum_{g'=1}^{G}\int_g \mathrm{d}E'\ \nu\sigma_f(\vec{r},E',t)\int \mathrm{d}\Omega'\ \psi(\vec{r},\hat{\Omega}',E',t) \\ + \int_g \mathrm{d}E\ q_e(\vec{r},\hat{\Omega},E,t),\end{aligned} \quad (1.13)$$

where v_g is the group velocity. Because the neutron velocity, v, is a continuous function of energy and the angular flux ψ is integrable in the same domain, then the first mean value theorem for integration allows

$$\int_g \mathrm{d}E\ \frac{1}{v(E)}\frac{\partial\psi}{\partial t} = \frac{1}{v(\xi)}\int_g \mathrm{d}E\ \frac{\partial\psi}{\partial t}, \quad (1.14)$$

where ξ is some energy within the group. Group cross-sections, fission spectrum, and fixed source can be defined as follows (Lewis & Miller [22]) to simplify the representation

$$\sigma_g(\vec{r},\hat{\Omega},t) = \frac{\int_g \mathrm{d}E\ \sigma(\vec{r},E,t)\psi(\vec{r},\hat{\Omega},E,t)}{\psi_g(\vec{r},\hat{\Omega},t)}, \quad (1.15)$$

$$\chi_g = \int_g \mathrm{d}E\ \chi(E), \quad (1.16)$$

$$\nu\sigma_{fg}(\vec{r},t) = \frac{\int_g \mathrm{d}E\ \nu\sigma_f(\vec{r},E,t)\int \mathrm{d}\Omega\ \psi(\vec{r},\hat{\Omega},E,t)}{\int \mathrm{d}\Omega\ '\psi_g(\vec{r},\hat{\Omega}',t)} \quad (1.17)$$

and

$$q_g^{\mathrm{e}}(\vec{r},\hat{\Omega},t) = \int_g \mathrm{d}E\ q_e(\vec{r},\hat{\Omega},E,t). \quad (1.18)$$

The multigroup group-to-group scattering cross section, σ_s, is considerably more complicated to define. The typical approach is to apply the discrete ordinates approximation, which yields

$$\sigma_{lgg'}(\vec{r},t) = \frac{\int_g \mathrm{d}E \ \int_{g'} \mathrm{d}E' \ \sigma_{sl}(\vec{r},E' \rightarrow E,t)\phi_{lm}(\vec{r},E',t)}{\phi_{lmg'}(\vec{r},t)}, \tag{1.19}$$

where

$$\phi_{lm}(\vec{r},E',t) = \int \mathrm{d}\Omega \ Y_{lm}(\hat{\Omega})\psi(\vec{r},\hat{\Omega},E',t) \tag{1.20}$$

and

$$\phi_{lmg'}(\vec{r},t) = \int \mathrm{d}\Omega \ Y_{lm}(\hat{\Omega})\psi_{g'}(\vec{r},\hat{\Omega},t). \tag{1.21}$$

While an angular dependence to the multigroup total cross section, σ_g, has been introduced, it can be eliminated by using a spherical harmonics expansion. Applying the discrete ordinates approximation, the multigroup form of the BTE is

$$\begin{aligned} &\left[\frac{1}{v_g}\frac{\partial}{\partial t} + \hat{\Omega}_n \cdot \vec{\nabla} + \sigma_g(\vec{r},t)\right]\psi_g(\vec{r},\hat{\Omega}_n,t) \\ &\qquad = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y^*_{lm}(\hat{\Omega}_n)\sum_{g'=1}^{G}\sigma_{lgg'}(\vec{r},t)\phi_{lmg'}(\vec{r},t) \\ &\qquad\qquad + \chi_g\sum_{g'=1}^{G}\nu\sigma_{fg'}\phi_{g'}(\vec{r},t) + q^{\mathrm{e}}_g(\vec{r},\hat{\Omega}_n,t), \end{aligned} \tag{1.22}$$

where ϕ_g is the group scalar flux. Segregating the within-group g from the scattering source yields the within-group BTE[1]

$$\begin{aligned} &\left[\frac{1}{v_g}\frac{\partial}{\partial t} + \hat{\Omega}_n \cdot \vec{\nabla} + \sigma_g(\vec{r},t)\right]\psi_g(\vec{r},\hat{\Omega}_n,t) \\ &\qquad = \sum_{l=0}^{L}(2l+1)P_l(\hat{\Omega}_n)\sigma_{lgg}(\vec{r},t)\phi_{lg}(\vec{r},t) + q_g(\vec{r},\hat{\Omega}_n,t), \end{aligned} \tag{1.23}$$

where q_g represents all the contributions to group g due to scatter from other energy groups (downscatter and upscatter), fission, and the fixed emitter.

[1] The within-group fission source is typically not segregated and is treated as a known value which is re-evaluated after the within-group solution is determined (Carlson & Lathrop [9]).

1.1.1 Source Iteration. The standard method for solving the within-group BTE is the scattering source iteration (SI), which is based on the von Neumann series solution. The angular flux, ψ, can be decomposed into the sum of collided angular flux contributions, $\tilde{\psi}^m$. The $\tilde{\psi}^m$ take on the physical significance that $\tilde{\psi}^0$ is the uncollided flux, $\tilde{\psi}^1$ is the one-collision flux, etc. We can define a series of iterates (Lewis & Miller [22]), such that

$$\begin{aligned} \psi^0 &= 0 \\ \psi^1 &= \tilde{\psi}^0 \\ &\vdots \\ \psi^{l+1} &= \sum_{m=0}^{l} \tilde{\psi}^m. \end{aligned} \tag{1.24}$$

Thus, ψ^{l+1} is the flux of neutrons that have had l or fewer collisions. A consequence of this method is that a problem that has a scattering-to-total cross section ratio, c, approaching unity requires an increasing number of iterations in order to converge to a solution. Furthermore, while the individual contribution to the angular flux from a specific iterate $\tilde{\psi}^m$, where m is large, might be small, the sum of the contributions from the high scatter iterates is not necessarily small. This can cause the source iteration method to falsely converge to a solution by terminating the iteration prematurely. In order to accelerate convergence to a solution, a variety of techniques are used, the most common one being diffusion synthetic acceleration (Alcouffe [3]). Using diffusion synthetic acceleration with source iteration is a well established method used extensively in production codes.

Diffusion synthetic acceleration generally performs well for a broad range of problems; however, it does lose effectiveness if the problem does not exhibit sufficient diffusive behavior. Furthermore, it is particularly difficult to implement the newer, better performing spatial quadratures that have been recently developed into diffusion synthetic acceleration. While other accelerators exist, none of them are a panacea.

1.1.2 Distribution Iteration. Distribution iteration (DI) offers several advantages over the more traditional source iteration method, most notably its rapid convergence when the scattering-to-total ratio approaches unity and its inherently parallel nature (Wager [32] and Prins [28]). The central idea behind DI is to reduce the global transport problem into a coupled-cell partial current problem that can be solved directly. The basic DI algorithm is shown in Algorithm 1.1. DI differs from source iteration in that it solves the contribution due to scattering directly and does not iterate on the scattering source. This key difference eliminates the need to use an accelerator to reduce the number of iterations required for convergence.

```
Guess the angular flux distribution on each face, e.g. isotropic
Compute the within group source due to fixed emission sources
repeat
  Use the angular flux distributions to determine the face-to-face transport probabilities
  Solve for the partial currents using the face-to-face transport probabilities
  Compute the within-group scattering source using the partial current solution and the angular flux distributions
  Refine the angular flux distributions
until the angular flux distributions are converged
Use the angular flux distributions and the partial currents to calculate the angular and scalar fluxes.
```

Algorithm 1.1: Overview of the distribution iteration algorithm.

1.2 Motivation

A reliable and robust time-dependent neutron transport code that can rapidly converge to a solution is useful for a variety of applications, one example being the determination of activation products in a fast-flux reactor. All current discrete ordinates implementations of time-dependent transport use the source iteration algorithm and exhibit poor performance for high scattering problems (Lewis & Miller [22] and Reed [29]).

Prior to my research, the distribution iteration algorithm had been implemented in slab and XY geometry for monoenergetic problems and isotropic emission sources.

For distribution iteration to be useful in a production code, it needed to be extended to support

- XYZ geometry
- multigroup transport
- time dependence

To have confidence that the distribution iteration algorithm is a suitable replacement for accelerated source iteration, I needed to:

- Implement a formal test plan that demonstrates that the distribution iteration algorithm correctly and reliably solves the discrete ordinates equations.
- To demonstrate the performance advantage that distribution iteration has over accelerated source iteration.

1.3 Statement of the Problem

The problem was to develop a robust time-dependent algorithm using the distribution iteration methodology and compare the performance relative to an established code, e.g. PARTISN. The extension from slab to XY geometry is complicated because:

- The angular quadratures are more complicated.
- The implementation of spatial quadratures, particularly a characteristic method, is more complicated.
- Mesh sweeps involve more than one pair of faces.
- The sparse matrix structure used to solve the partial current problem has a more complex structure.

The extension from XY to XYZ geometry is conceptually simpler than the extension from slab to XY geometry because the angular quadratures are similar, the spatial quadratures are derived in a similar fashion, and the complexity of the mesh sweeps and the sparse matrix are comparable. However, a “just one more axis” approach based on the work done by Prins [28] would make a complicated implementation into an unmanageable software engineering problem and, hence, difficult to verify.

The DI algorithm as developed by Wager and Prins sometimes failed to converge when only one or two inner iterations were used to refine the angular current distribution per call to the partial current solver. Prins was able to achieve convergence for his test problems by performing up to ten refinement iterations per call to the partial current solver. Though this method did improve convergence, it was not clear that it would work in all cases and it was an inefficient approach. Prins attempted several other approaches to improve the refinement of the angular current distribution, however, none of them were completely successful.

1.4 Goals of the Research

The work done by Wager [32] and Prins [28] demonstrated the robust performance of DI for time-independent problems in slab and XY geometries. The conventional algorithms for solving multiple energy groups and time dependent problems work by modifying the emission density, q, and the total cross section, σ. The use of DI with these algorithms has not been previously demonstrated and it is uncertain whether the changes to the emission density and the total cross section will adversely impact the performance of the DI algorithm.

The goals of this research were:

- Solving the time-dependent BTE using DI.
- Extending the time-dependent DI methodology to multiple energy groups.
- Extending DI to support XYZ geometry in a straightforward and methodical manner.
- Improving the performance of the angular current distribution refinement algorithm.

- Demonstrating that DI is a viable substitute for source iteration.

The ultimate goal of my research was to build upon the previous work and demonstrate that DI is a viable alternative to source iteration, either unaccelerated or accelerated, for a broad range of transport problems.

1.5 Scope

To support the goals of my research, a multigroup, time-dependent transport code utilizing DI was created. The code supports slab, XY and XYZ geometries using a variety of spatial and angular quadratures. Time-dependent external sources are supported and there is limited support for time-dependent material properties. The formal verification effort was focused on slab geometry and XY geometry. The verification of the XYZ geometry implementation was designed to take advantage of common design basis between the XY and XYZ geometry implementations.

The verification testing was designed with two separate requirements. The first requirement was to exercise specific aspects of the DI algorithm in order to check for error in the corresponding code segments. The second requirement was to test the parameter space, e.g. scattering ratio, in order to demonstrate that DI correctly and robustly solves the discrete ordinates equations.

The data collected during the verification testing was then used to evaluate the performance of both the distribution iteration and source iteration algorithms.

1.6 Assumptions and Limitations

Isotropic cross-sections are used in order to focus the research on the implementation of the DI method to time-dependent transport. While implementing higher-order terms to support anisotropic scatter would not be difficult, it does add complexity to the software and complicate verification testing.

The time-dependent material properties, such as density and composition, were held constant during testing. The research did not address the implicit relationship between neutron flux and material properties. The software, however, was designed to allow for changes in material properties.

The multigroup energy support in my research was limited to non-fissioning materials. While adding support for fissioning problems is not complicated algorithmically, the effort required was better utilized towards the other research goals. A demonstration of multigroup support based on up and downscatter is more than sufficient to show the applicability of distribution iteration as a replacement for source iteration.

The performance metric was limited to iteration count rather than run time because iteration count provides an indication of the efficiency of the code. Run time is a subjective assessment of a code because design requirements may result in different performance trade-offs.

1.7 Approach

The first step was to determine a mathematical formulation of the time-dependent BTE that was amenable to the distribution iteration approach and maintains a close connection to the physics. Next, a one-dimensional, time-dependent distribution iteration was implemented for monoenergetic problems and the performance of that code was compared relative to PARTISN. Finally, the development of metrics for comparing the performance relative to a benchmark was undertaken. Some of the points of investigation were to:

- Evaluate the application of more robust spatial quadratures, such as first moment and characteristic methods.
- Evaluate time stepping methods.
- Evaluate methods to improve the propagation of global neutron partial currents.

The first version of the time-dependent DI code (DI-1) highlighted the requirement for a more effective algorithm for propagating global neutron partial currents in deep-penetration problems. Addressing that issue was undertaken during the development of the XY geometry version of the code (DI-2).

The implementation of an improved time-stepping method was the next phase of the research. The DI-1 code developed to support my research prospectus was implemented using an implicit backward difference method to numerically approximate the time derivative. While this approach was straightforward, truncation error affected the performance. The most obvious approach was to implement the midpoint average method in lieu of the backward difference method utilized by DI-1. The second version, DI-2, implemented both time stepping methods.

The implementation of XYZ geometry was accomplished in the third version (DI-3) of my transport code. The final phase was the extension of the code to support multiple energy groups. This was the last phase because it is dependent on having a robust monoenergetic implementation.

II. Theory

Numerical methods for solving time-dependent (evolutionary) partial differential equations can be characterized by how the time variable is treated and how the extrapolation to a future state is performed. If all the derivatives–both spatial and time–are represented with finite differences, then the method is fully discretized (Iserles [17]). For example, given

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \tag{2.1}$$

then a fully discretized scheme using the forward Euler method can be expressed as

$$u_l^{j+1} = u_l^j + \frac{\Delta t}{(\Delta x)^2}\left(u_{l-1}^j - 2u_l^j + u_{l+1}^j\right), \tag{2.2}$$

where $\Delta t/(\Delta x)^2$ is the Courant number, and u is a vector of spatial values indexed by l at times j and $j+1$. An alternative method is the semi-discretization method. In this formulation a partial differential equation is transformed into a coupled system of ordinary differential equations. My research is based on fully discretized methods.

The means of extrapolating to the next time step can be categorized as either explicit or implicit. An explicit method, such as Euler's method, only utilizes values from the current time step to compute values for the next time step. For example, given

$$\frac{\mathrm{d}u}{\mathrm{d}t} = f(u,t) \tag{2.3}$$

the time derivative is approximated by using a truncated Taylor series expansion as

$$\left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t^j} = \frac{u^{j+1} - u^j}{\Delta t}. \tag{2.4}$$

The differential equation can be solved for each time step by

$$u^{j+1} = u^j + \Delta t f(u^j, t^j). \tag{2.5}$$

Because the error term in the truncated Taylor series expansion is $O(\Delta t^2)$, the global truncation error is first order. While the forward Euler method is convergent (Iserles [17]), it is only conditionally stable. An alternative explicit method is the explicit midpoint method defined as

$$u^{j+2} = u^j + 2\Delta t f(u^{j+1}, t^{j+1}). \tag{2.6}$$

The explicit midpoint method is classified as a multistep method and is second order.

An implicit method, such as the trapezoidal rule, utilizes values from multiple time steps, including extrapolated values, to compute values for the next time step. For example, for the backward Euler method, the time derivative is approximated as

$$\frac{\mathrm{d}u}{\mathrm{d}t}\bigg|_{t^{j+1}} = \frac{u^{j+1} - u^j}{\Delta t}. \tag{2.7}$$

Thus, the differential equation is solved by

$$u^{j+1} = u^j + \Delta t f(u^{j+1}, t^{j+1}). \tag{2.8}$$

The backward Euler method has the same local truncation error as the forward Euler method and is also convergent. Unlike the forward Euler method, the backward Euler method is unconditionally stable, which is a significant advantage when dealing with stiff problems (Iserles [17]).

The primary disadvantage to using the backward Euler method is the first order global truncation error. The trapezoidal rule,

$$u^{j+1} = u^j + \frac{\Delta t}{2}\left(f(u^{j+1}, t^{j+1}) - f(u^j, t^j)\right), \tag{2.9}$$

has second order global truncation error (Iserles [17]), is convergent and is unconditionally stable. The disadvantage to using the trapezoidal rule to solve the BTE is the additional memory requirement for storing $f(u^{j+1}, t^{j+1})$ and $f(u^j, t^j)$. The implicit midpoint method evaluates f at the midpoint between the steps j and $j+1$ and yields

$$u^{j+1} = u^j + \Delta t f\left(\frac{u^{j+1} + u^j}{2}, t^{j+1/2}\right) \tag{2.10}$$

and also has second order local truncation error.

An implicit method can be solved by making it into a semi-implicit method, which means the method has been solved by linearization. The strategy behind a semi-implicit method is to split the problem into two portions, one that will be solved implicitly and another that will be solved explicitly. An example of a semi-implicit method is PARTISN's approach to solving the time-dependent BTE, which is discussed in detail in section 2.4.4 and Appendix A. A simple illustration of a semi-implicit method can be shown by solving the backward Euler method for f,

$$f(u^{j+1}, t^{j+1}) = \frac{u^{j+1} - u^j}{\Delta t}. \tag{2.11}$$

Substituting for f into the forward Euler method yields

$$u^{j+2} = 2u^{j+1} - u^j. \tag{2.12}$$

Performing a Taylor series expansion of u^{j+2} and u^j about u^{j+1} yields

$$u^{j+2} = u^{j+1} + \Delta t f(u^{j+1}, t^{j+1}) + \frac{(\Delta t)^2}{2} f'(u^{j+1}, t^{j+1}) + O((\Delta t)^3) \tag{2.13}$$

and

$$u^j = u^{j+1} - \Delta t f(u^{j+1}, t^{j+1}) + \frac{(\Delta t)^2}{2} f'(u^{j+1}, t^{j+1}) - O((\Delta t)^3). \tag{2.14}$$

Taking the difference between u^{j+2} and u^j yields

$$u^{j+2} - u^j = 2\Delta t f(u^{j+1}, t^{j+1}) + O((\Delta t)^3), \tag{2.15}$$

which is the explicit midpoint method and is second order.

Before delving into the implementation of the time-dependent BTE in DI, it is useful to cover some background involving the time-independent BTE.

2.1 Operator Notation

For clarity, the BTE can be represented in operator form as

$$\mathbb{V}^{-1}\frac{\partial}{\partial t}\Psi = -B\Psi + Q^{\mathrm{e}} \tag{2.16}$$

where $\mathbb{V}$ is a diagonal matrix of neutron group velocities, Ψ is a vector containing angular fluxes for each group, and Q^{e} is a vector of embedded sources. The transport operator B is

$$B\Psi = (L - S)\Psi \tag{2.17}$$

$$(L\Psi)_g = \hat{\Omega} \cdot \nabla\Psi_g + \sigma_{\mathrm{t}}(r,t)\Psi_g \tag{2.18}$$

$$(S\Psi)_g = \sum_{g'} \int \mathrm{d}\hat{\Omega}'\ K_{g' \to g}(\vec{r}, \hat{\Omega}' \to \hat{\Omega})\Psi_{g'}(\vec{r}, \hat{\Omega}'), \tag{2.19}$$

where K represents the scattering kernel and σ_{t} is the total cross section. The operator L represents losses and the operator S represents scattering sources.

2.2 Spatial Quadratures

Many spatial quadratures have been developed for use with the discrete ordinates formulation of the BTE. The spatial quadratures can be broadly grouped as zeroth moment and first moment methods. The zeroth moment methods only use cell average values, e.g., cell average angular fluxes, while the first moment methods use both the cell average values and the first spatial Legendre moments.

The most common zeroth moment methods are the Diamond Difference (DD), Step Characteristic (SC) (Lathrop [20]), and Step approximations. The diamond difference spatial quadrature uses the auxiliary relationship (in slab geometry)

$$\psi^{\mathrm{L}} + \psi^{\mathrm{R}} = 2\psi^{\mathrm{A}}, \tag{2.20}$$

where ψ^{L} and ψ^{R} are the angular fluxes on the left and right boundaries, respectively. The advantage to using diamond difference is the simplicity of its implementation and second-order accuracy. The primary disadvantage is the fact that it will generate physically meaningless negative fluxes in some circumstances. The step spatial quadrature uses the auxiliary relationships

$$\psi^{\mathrm{R}} = \psi^{\mathrm{A}} \quad \mu > 0 \tag{2.21}$$

$$\psi^{\mathrm{L}} = \psi^{\mathrm{A}} \quad \mu < 0. \tag{2.22}$$

The step spatial quadrature is simpler than diamond difference, however, it only has first-order convergence, which makes it impractical to use in a production code. It is, however, a "positive" method in the sense that it will always generate positive fluxes. The step characteristic spatial quadrature is based on the assumption that particles travel in straight lines between collisions. These straight lines are the characteristics of the BTE and are represented by

$$\frac{\mathrm{d}\psi}{\mathrm{d}s} + \sigma_{\mathrm{t}}\psi = q, \tag{2.23}$$

where s is the path traveled by the particle. Without loss of generality, consider the $\mu > 0$ case. Integrating the characteristic equation yields

$$\psi(s) = \psi^{\mathrm{L}} e^{-\sigma_{\mathrm{t}} s} + \int_0^s \mathrm{d}s' \; q(s') e^{-\sigma_{\mathrm{t}}(s-s')}. \tag{2.24}$$

The step characteristic spatial quadrature has the simplifying approximation that $q(s) = q^A$, where q^A is the cell average source. The advantage to using SC is that it is a positive method and has second-order convergence. The drawback to SC is that it is difficult to generalize to non-rectangular coordinate systems.

The most common first moment methods are Linear Discontinuous (LD) (Hill [15]) and Linear Characteristic (LC) (Larsen & Alcouffe [18]). The first moment methods have the drawback of increased memory requirements, however, they do exhibit a better rate of convergence than the zeroth moment methods. The linear discontinuous spatial quadrature assumes that the flux is linear within a cell but discontinuous with the flux in adjacent upstream cells. This method can produce negative fluxes with optically thick cells. The linear characteristic method is similar to the step characteristic method with a first-moment expansion for the source

$$q(x) = q^A P_0(x) + q^X P_1(x), \tag{2.25}$$

where q^X is the first moment and P_n are the Legendre polynomials shifted and scaled in x. Though LD is not a positive method, it produces fewer negative fluxes than DD.

There are new spatial quadratures that have been developed which have not gained widespread acceptance. An important reason for the lack of acceptance is due to the complexity of developing an accelerator for use with source iteration. Some examples of the newer spatial quadratures are Exponential Discontinuous (Wareing [34]), Nonlinear characteristic (Walters & Wareing [33]), Exponential Characteristic (Mathews, Sjoden & Minor [25]) and Nonlinear Corner Balance (Castrianni & Adams [10])

The negative fluxes can have a deleterious effect on the performance of a transport code (Lathrop [20]). In multidimensional problems, the negative fluxes can slow the rate of convergence or render the acceleration method used with source iteration ineffective. The propagation of negative fluxes can result in a lack of stability for time dependent problems. Traditionally, a negative flux fix-up is typically used with non-positive methods like DD. The drawback to using a negative flux fix-up is a slower rate of convergence.

2.3 Distribution Iteration

Distribution iteration solves the BTE by solving two different aspects: the partial current problem and the distribution of the angular partial currents on the cell faces. The partial currents problem is a simplification of the transport problem into a linear algebra problem that can be solved directly. The second aspect, solving for the distribution of the angular partial currents, serves two purposes–providing the means for improving the partial currents problem and, second, providing the solution to the angular and scalar fluxes in each cell.

Before proceeding, a brief explanation on notation will help guide the reader through the derivations that follow. The use of the vector decoration, e.g., $\vec{J}$, denotes a vector constructed from multiple faces. Symbols which are inherently vectors because of the use of an angular quadrature, such as the cell average angular flux (ψ^{A}), are not decorated. The rationale for restricting the use of the vector decoration was to make it clear to the reader which variables are constructed from multiple faces. The use of the“blackboard bold” style, e.g. $\mathbb{M}$, denotes a matrix. I have also adopted the convention of using the “e” superscript to denote all external contributions to the within-group BTE.

The DI algorithm models the BTE as a collection of cells coupled together by partial current flows of neutrons between the cells. An implicit representation of the global partial currents problem can be expressed as (Mathews [24])

$$\vec{J} = \vec{J}^{\mathrm{e}} + \mathbb{M}\vec{J}, \tag{2.26}$$

where $\vec{J}$ is the vector form of all the partial currents across cell faces; $\vec{J}^{\mathrm{e}}$ is the emission partial current due to scatter from other groups, fission, and fixed emitters; and $\mathbb{M}$ is a matrix of the mean face-to-face transport probabilities.

The advantage of (2.26) is that $\vec{J}$ is non-negative because each element of $\mathbb{M}$ is non-negative, as is each element of $\vec{J}^{\mathrm{e}}$. The implicit form does have the disadvantage that an iterative solution will exhibit slow convergence for optically thick cells as $c \rightarrow 1$ (Mathews [24]). Solving for $\vec{J}$ explicitly yields

$$\vec{J} = (\mathbb{I} - \mathbb{M})^{-1}\vec{J}^{\mathrm{e}}, \tag{2.27}$$

where $\mathbb{I}$ is the identity matrix. If we apply synthetic division to the matrix inversion, we have

$$(\mathbb{I} - \mathbb{M})^{-1} = 1 + \mathbb{M} + \mathbb{M}^2 + ..., \tag{2.28}$$

thus the inverted matrix, which we identify as $\mathbb{G}$, represents all possible transport paths that a neutron might take over the entire spatial domain.

A method for determining the elements of the matrix $\mathbb{M}$ is required. The first step is to formulate the equations for within-cell transport[1] as

$$\psi^{\mathrm{A}} = \mathbb{K}^{\psi \mathrm{I}}\vec{j}^{\mathrm{in}} + \mathbb{K}^{\psi \mathrm{E}}q^{\mathrm{e}} + \mathbb{K}^{\psi \mathrm{S}}\Sigma\psi^{\mathrm{A}} \tag{2.29}$$

and

$$\vec{j}^{\mathrm{out}} = \mathbb{K}^{\mathrm{OI}}\vec{j}^{\mathrm{in}} + \mathbb{K}^{\mathrm{OE}}q^{\mathrm{e}} + \mathbb{K}^{\mathrm{OS}}\Sigma\psi^{\mathrm{A}}, \tag{2.30}$$

where ψ^{A} is the cell average angular flux, $\vec{j}^{\mathrm{in}}$ is the inward flowing currents organized on a per-cell basis and $\vec{j}^{\mathrm{out}}$ is the outward flowing currents organized on a per-cell basis. The physical significance of each matrix is

- $\mathbb{K}^{\mathrm{OI}}$ accounts for the contribution from the uncollided neutrons to the cell outflow,
- $\mathbb{K}^{\mathrm{OE}}$ accounts for the contribution from the cell fixed emission source to the cell outflow,

[1] I have adopted an angular partial current notation vice an angular flux notation.

- $\mathbb{K}^{\mathrm{OS}}$ accounts for the contribution from the cell scattering source to the cell outflow,
- $\mathbb{K}^{\psi\mathrm{I}}$ accounts for the contribution from the uncollided neutrons to the cell average angular flux,
- $\mathbb{K}^{\psi\mathrm{E}}$ accounts for the contribution from the cell fixed emission source to the cell average angular flux,
- $\mathbb{K}^{\psi\mathrm{S}}$ accounts for the contribution from the cell scattering source to the cell average angular flux, and
- Σ accounts for the scatters within the cell.

The $\mathbb{K}$ matrices are dependent upon the spatial quadrature scheme, angular quadrature, and material properties.

Solving (2.29) for ψ^{A} yields

$$\left(\mathbb{I} - \mathbb{K}^{\psi\mathrm{S}}\Sigma\right)\psi^{\mathrm{A}} = \mathbb{K}^{\psi\mathrm{I}}\vec{\jmath}^{\,\mathrm{in}} + \mathbb{K}^{\psi\mathrm{E}}q^{\mathrm{e}}, \tag{2.31}$$

where $\mathbb{I}$ is the identity matrix. Let

$$\mathbb{L} = \left(\mathbb{I} - \mathbb{K}^{\psi\mathrm{S}}\Sigma\right)^{-1}. \tag{2.32}$$

Applying synthetic division to the matrix inversion yields

$$\mathbb{L} = \mathbb{I} + \mathbb{K}^{\psi\mathrm{S}}\Sigma + \left(\mathbb{K}^{\psi\mathrm{S}}\Sigma\right)^2 + ... \tag{2.33}$$

Thus, the matrix $\mathbb{L}$ accounts for all combinations of transport within a cell–uncollided, single collision, two collisions, etc.–and the cumulative contribution to the cell average angular flux. Substituting in $\mathbb{L}$ yields

$$\psi^{\mathrm{A}} = \mathbb{L}\mathbb{K}^{\psi\mathrm{I}}\vec{\jmath}^{\,\mathrm{in}} + \mathbb{L}\mathbb{K}^{\psi\mathrm{E}}q^{\mathrm{e}}. \tag{2.34}$$

Substituting (2.34) into (2.30) yields

$$\vec{j}^{\mathrm{out}} = \left(\mathbb{K}^{\mathrm{OI}} + \mathbb{K}^{\mathrm{OS}}\Sigma\mathbb{L}\mathbb{K}^{\psi\mathrm{I}}\right)\vec{j}^{\mathrm{in}} + \left(\mathbb{K}^{\mathrm{OE}} + \mathbb{K}^{\mathrm{OS}}\Sigma\mathbb{L}\mathbb{K}^{\psi\mathrm{E}}\right) q^{\mathrm{e}}. \tag{2.35}$$

The coefficient matrix and external angular partial current vector are defined as

$$\mathbf{m}^{\mathrm{oi}} = \mathbb{K}^{\mathrm{OI}} + \mathbb{K}^{\mathrm{OS}}\Sigma\mathbb{L}\mathbb{K}^{\psi\mathrm{I}} \tag{2.36}$$

and

$$\vec{j}^{\mathrm{e}} = \left(\mathbb{K}^{\mathrm{OE}} + \mathbb{K}^{\mathrm{OS}}\Sigma\mathbb{L}\mathbb{K}^{\psi\mathrm{E}}\right) q^{\mathrm{e}}. \tag{2.37}$$

The partial currents vector $\vec{J}^{\mathrm{e}}$ is computed from the angular partial currents by using the angular quadrature weights. The angular quadrature weight matrix, $\mathbb{W}$, is constructed such that

$$\vec{J}^{\mathrm{out}} = \mathbb{W}\vec{j}^{\mathrm{out}}. \tag{2.38}$$

The equation for the partial current for each cell is then

$$\vec{J} = \mathbb{W}\mathbf{m}^{\mathrm{oi}}\vec{j}^{\mathrm{in}} + \mathbb{W}\vec{j}^{\mathrm{e}}. \tag{2.39}$$

The inward flux distribution, ζ, is defined as

$$\zeta_{face}(\vec{r},\hat{\Omega}) = \frac{\left|\hat{\Omega}\cdot\hat{n}_{face}^{\mathrm{outward}}\right| \psi(\vec{r},\hat{\Omega})}{\int_{\hat{r}'\in\Gamma_{face}} \mathrm{d}\Gamma' \int_{\hat{\Omega}'\cdot\hat{n}_{face}^{\mathrm{outward}}<0} \mathrm{d}\Omega' \left|\hat{\Omega}'\cdot\hat{n}_{face}^{\mathrm{outward}}\right| \psi(\vec{r}',\hat{\Omega}')}, \tag{2.40}$$

where $\hat{n}_{face}^{\mathrm{outward}}$ is the outward normal on a cell face $face$. The flux distribution ζ apportions the partial current on a cell face to the individual ordinates that constitute the inward flow into a cell. The inward flux distribution can be represented in matrix form as

$$\mathbb{Z}_{cell,face} = \frac{\vec{j}^{\mathrm{in}}_{cell,face}}{J_{cell,face}}, \tag{2.41}$$

where $\mathbb{Z}_{cell,face}$ is a subset of $\mathbb{Z}$ that, for a given cell and face, $\vec{j}^{\text{in}}_{cell,face}$ is the portion of $\vec{j}^{\text{in}}$ for a cell that corresponds to $face$ and $J_{cell,face}$ is the inward partial current for a cell that corresponds to $face$, which is a scalar value. Using $\mathbb{Z}$ to eliminate $\vec{j}^{\text{in}}$ yields

$$\vec{J} = \mathbb{W}\mathbf{m}^{\text{oi}}\mathbb{Z}\vec{J} + \mathbb{W}\vec{j}^{\text{out}}_{e}. \tag{2.42}$$

Thus, we can determine $\mathbb{M}$ from

$$\mathbb{M} = \mathbb{W}\mathbf{m}^{\text{oi}}\mathbb{Z} \tag{2.43}$$

and $\vec{J}^{\text{e}}$ from

$$\vec{J}^{\text{e}} = \mathbb{W}\vec{j}^{\text{e}}. \tag{2.44}$$

The basic algorithm for DI that I implemented in my research is summarized in Algorithm 2.1. There are two key points in the algorithm that differ from the implementations by Wager and Prins that I will briefly highlight. The most significant point of departure is in the sweeping methodology. The cell outward angular partial currents can be determined from the equation

$$\vec{j}^{\text{out}} = \mathbb{K}^{\text{OI}}\vec{j}^{\text{in}} + \mathbb{K}^{\text{OS}}\Sigma\mathbb{L}\mathbb{K}^{\psi\text{I}}\vec{j}^{\text{in}} + \vec{j}^{\text{e}}. \tag{2.45}$$

The cell outflow due to within-cell sources is defined as

$$\vec{j}^{\text{os}} = \mathbb{A}'\vec{j}^{\text{in}} + \vec{j}^{\text{e}}, \tag{2.46}$$

where

$$\mathbb{A}' = \mathbb{K}^{\text{OS}}\Sigma\mathbb{L}\mathbb{K}^{\psi\text{I}}. \tag{2.47}$$

This change to the algorithm solves the stability problem that was inherent in the previous algorithm. The algorithm used by Wager and Prins would update the scattering source while sweeping while this algorithm computes the scattering source for all cells before sweeping. The Wager and Prins algorithm uses partially updated cell inflows when calculating the scattering source, which can cause unstable behavior, particularly in XY and XYZ geometry. My algorithm also has improved parallel efficiency because each cell can be calculated independently of the others. Also, the contribution due to sources in each cell can be computed for all ordinates in a cell using a sparse BLAS library.

The other point that bears further explanation is the phrase "Until $\mathbb{Z}$ stabilizes." Typically, only one iteration is needed to refine the inward current distribution. During the course of my research, there were some configurations that required more than one iteration in order for DI to converge, but two always sufficed (in my testing). Alternatively, one can iterate until the inward current distribution converges. In fact, it is possible to solve the transport equation this way without explicitly solving for the partial currents, but the "no partial current solver" is a poor algorithm: The overall number of iterations increases because the refinement of the inward current distribution does not propagate the cell-to-cell transport efficiently. There is, however, an optimization between the number of iterations required to refine the inward current distribution in order to converge and the relative CPU cost between solving the partial currents problem and refining the inward current distribution.

Because the computation of the angular flux distribution within the cell can be performed independently of any other cell, parallelization can be accomplished with modest effort. Furthermore, parallel implementations of linear solvers are well established, which benefits the solution of the global partial currents problem.

2.3.1 Partial Currents in Slab Geometry. Before discussing the partial currents in XYZ geometry, an example from slab geometry illustrates the basic terminology. Figure 2.1 is a simple example illustrating the coupling between two slabs. Let J_i^+ and J_i^- represent the partial currents in the positive μ and negative μ directions for face i, respectively.

```
Compute transport matrices (e.g. $\mathbb{K}$, $\Sigma$, and $\mathbb{W}$)
Initialize inward current distribution, $\mathbb{Z}$ (e.g. isotropic)
Compute within group source ($\vec{j}^{\mathrm{e}}$ and $\vec{J}^{\mathrm{e}}$)
repeat
  Compute $\mathbb{M}$
  Solve $(\mathbb{I} - \mathbb{M})\vec{J} = \vec{J}^{\mathrm{e}}$
  repeat
    Compute the source outflow in each cell as $\vec{j}^{\mathrm{os}} = \vec{j}^{\mathrm{e}} + \mathbb{A}'\mathbb{Z}\vec{J}$
    for each octant do
      Initialize inward angular current, $\vec{j}^{\mathrm{in}}$ from boundary
      for each cell in flow sequence do
        Compute outward cell angular partial currents $\vec{j}^{\mathrm{out}} = \mathbb{K}^{\mathrm{OI}}\vec{j}^{\mathrm{in}} + \vec{j}^{\mathrm{os}}$
        Propagate outward cell angular partial currents to neighboring cells as
        new inward cell angular partial currents
      end for
    end for
    Update $\mathbb{Z}$ using updated inward cell angular partial currents
  until $\mathbb{Z}$ stabilizes
until $\mathbb{Z}$ converges
```

Algorithm 2.1: Distribution Iteration Algorithm

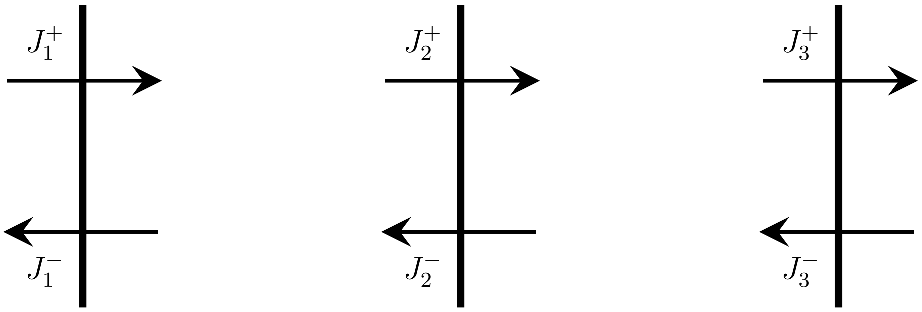


Figure 2.1: Coupled Cells in Slab Geometry

The partial currents shown in Figure 2.1 can be expressed as

$$J_1^- = M_{1,1}^- J_1^+ + M_{1,2}^- J_2^- + J_1^{e-} \tag{2.48}$$

$$J_1^+ = M_{1,1}^+ J_1^- + J_1^{e+} \tag{2.49}$$

$$J_2^- = M_{2,2}^- J_2^+ + M_{2,3}^- J_3^- + J_2^{e-} \tag{2.50}$$

$$J_2^+ = M_{2,1}^+ J_1^+ + M_{2,2}^+ J_2^- + J_2^{e+} \tag{2.51}$$

$$J_3^- = M_{3,3}^- J_3^+ + J_3^{e-} \tag{2.52}$$

$$J_3^+ = M_{3,2}^+ J_2^+ + M_{3,3}^+ J_3^- + J_3^{e+} \tag{2.53}$$

using the notation $M_{\text{to,from}}^{\text{direction}}$ to represent the mean probability that a neutron will travel from face $from$ to face to and $J_{\text{face}}^{e\pm}$ is the fixed source current on a face in either a positive or negative μ direction. When constructing the current vector $\vec{J}$ from the individual $J_i^\pm$, I adopted the following convention:

$$\vec{J} = \begin{bmatrix} J_1^- \\ J_1^+ \\ J_2^- \\ J_2^+ \\ \vdots \end{bmatrix}. \tag{2.54}$$

The individual m's are used to construct a linear system that solves for the $\vec{J}$ current vector, specifically $(\mathbb{I} - \mathbb{M})\vec{J} = \vec{J}^e$, where

$$\mathbb{M} = \begin{bmatrix} 0 & M_{1,1}^- & M_{1,2}^- & 0 & 0 & 0 & ... \\ M_{1,1}^+ & 0 & 0 & 0 & 0 & 0 & ... \\ 0 & 0 & 0 & M_{2,2}^- & M_{2,3}^- & 0 & ... \\ 0 & M_{2,1}^+ & M_{2,2}^+ & 0 & 0 & 0 & ... \\ & & & \vdots & & & \end{bmatrix}. \tag{2.55}$$

2.3.2 Partial Currents in XYZ Geometry. Prins [28] developed an XY geometry implementation of DI. Conceptually, the partial currents problem in XY geometry can be considered the same as four slab geometry partial currents. The four arrangements are the Left/Right coupling, the Top/Bottom coupling, and the two diagonal couplings, Left/Bottom to Right/Top and Right/Bottom to Left/Top. Extending into XYZ geometry involves extending those couplings into the Z direction.

By deconstructing the partial currents and treating all the cell faces identically, a simple arrangement of cell faces into one vector is possible. One of my implementation goals was to maximize the common code elements between the different geometries. That goal leads to an extensible arrangement of the partial currents vector where all the faces normal to the X axis (X faces) are grouped together, all the faces normal to the Y axis (Y faces) are grouped together, and all the faces normal to the Z axis (Z faces) are grouped together. In a slab geometry problem of N_x cells, there are $N_\mathrm{x} + 1$ faces. In XY geometry of N_x by N_y cells, there will be $(N_\mathrm{x} + 1)N_\mathrm{y}$ X faces and $(N_\mathrm{y} + 1)N_\mathrm{x}$ Y faces. The partial current vector would then consist of a vector containing a block of X faces and block of Y faces. In XYZ geometry of N_x by N_y by N_z cells, there will be $(N_\mathrm{x}+1)N_\mathrm{y}N_\mathrm{z}$ X faces, $(N_\mathrm{y}+1)N_\mathrm{x}N_\mathrm{z}$ Y faces and $(N_\mathrm{z}+1)N_\mathrm{x}N_\mathrm{y}$ Z faces. The partial current vector would be extended by a block of Z faces.

This arrangement of partial currents leads to an interesting arrangement of the matrix $\mathbb{I} - \mathbb{M}$. The terms adjacent to the main diagonal are associated with the axial Left/Right, Bottom/Top, and Back/Front transport. Off the diagonal are clusters that are associated with the diagonal transport, e.g., Left/Top. Figure 2.2 shows a graphical representation of the $\mathbb{G}$ matrix. The columns are labeled with the associated inflow dimension and the rows are labeled with the associated outflow dimension.

2.3.3 Transport Coefficients in XYZ Geometry. Appendix B shows the derivation of the transport coefficients for the step and diamond-difference spatial quadratures in XYZ geometry. For diamond-difference to have positive fluxes, all the transport coefficients must be positive. All the coefficients, except for the three axial

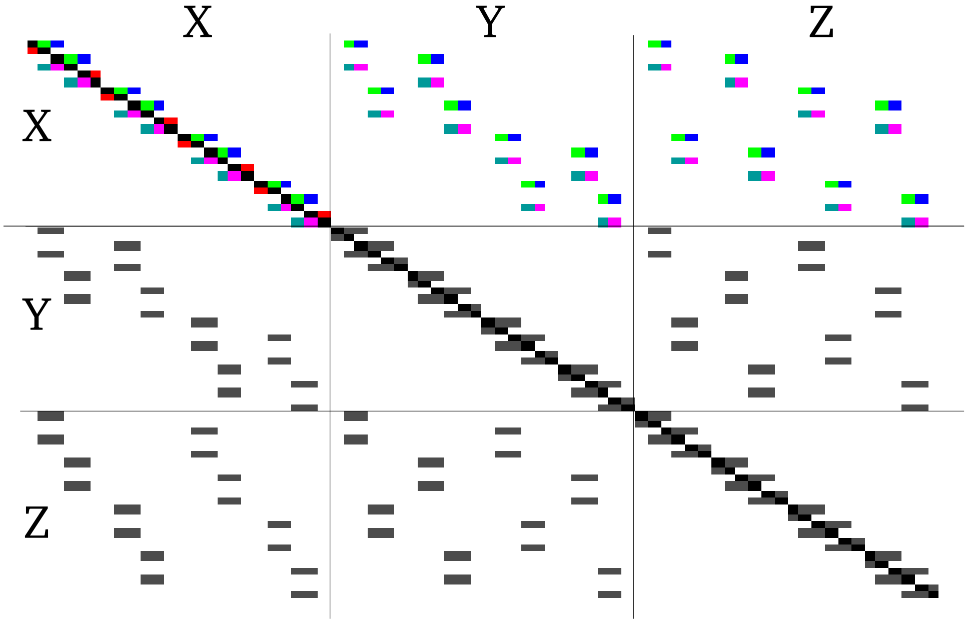


Figure 2.2: Graphical Representation of the Global Partial Currents Matrix

transmission coefficients, are unconditionally positive. The three axial transmission coefficients, $\mathbb{K}_{xx}^{\mathrm{OI}}$, $\mathbb{K}_{yy}^{\mathrm{OI}}$ and $\mathbb{K}_{zz}^{\mathrm{OI}}$, are positive if

$$(\alpha - \beta) - 1 > \frac{\epsilon_y}{2}, \tag{2.56}$$

$$1 - (\alpha + \beta) > \frac{\epsilon_y}{2} \tag{2.57}$$

and

$$(\beta - \alpha) - 1 > \frac{\epsilon_y}{2}, \tag{2.58}$$

respectively, are satisfied. However, it is impossible for all three to hold. In fact, at most only one can be satisfied. In the S$_n$ angular quadrature, the positive coefficients will be the ones closest to the axis. For example, if $\mu > \eta + \xi$ for an ordinate, its $\mathbb{K}_{xx}^{\mathrm{OI}}$ is positive and its $\mathbb{K}_{yy}^{\mathrm{OI}}$ and $\mathbb{K}_{zz}^{\mathrm{OI}}$ are negative.

The consequence of negative coefficients is that the angular flux, in the absence of sufficiently strong within-cell sources, can be biased towards axial flows. This effect is not unique to DI, it will also occur in the source iteration algorithm. Thus, in XYZ geometry, the use of diamond-difference can result in anisotropic flux distributions as an artifact of the spatial quadrature.

In the DI algorithm, negative fluxes can cause problems with the inward angular flux distribution. If some of the elements of $\vec{j}^{\text{in}}$ are negative and the inward partial current remains positive, then ζ will be negative where $\vec{j}^{\text{in}}$ is negative. However, if enough elements of $\vec{j}^{\text{in}}$ are negative such that the partial current is negative, then ζ will have the opposite polarity with respect to $\vec{j}^{\text{in}}$. The sign of ζ also affects the sign of the elements of $\mathbb{M}$, which can result in the additional production of negative partial currents. As a result, the elements of the matrix $\mathbb{Z}$ can oscillate and not converge.

The conventional method for implementing a negative flux fix-up is not the best approach in DI because it requires additional storage or more computations. Instead, I developed a straightforward negative flux fix-up scheme that takes advantage of the normalization of ζ. During a mesh sweep, if a negative current is generated, it is set to zero. When the matrix $\mathbb{Z}$ is updated, the corrected angular partial current will not contribute to the partial current, thus particle conservation is assured.

2.4 Time Discretization

The most established time dependent BTE transport codes are the 1972 TIMEX code (Reed [30]), the 1976 TIMEX code (Hill [16]) and PARTISN (Alcouffe [4]). Some of the other time-dependent codes that have been developed are TDTORT (Goluoglu [14]), TDA (Time-Dependent ANISN) (Dupree [11]), and TRANZIT (Lathrop [21]).

2.4.1 TIMEX (1972) Implementation. The TIMEX (1972) code (Reed [30]) utilizes a first order Taylor series expansion of the time derivative to discretize the time variable analogous to forward Euler. TIMEX (1972) also uses the multigroup approximation to discretize energy and the discrete ordinates representation to discretize in angle. The BTE solved by TIMEX (1972) in slab geometry (Appendix A) in operator notation is

$$\mathbb{V}^{-1}\frac{\partial}{\partial t}\Psi^{j+1} + L\Psi^{j+1} = S\Psi^{j} + Q^{\mathrm{e}}. \tag{2.59}$$

Of particular interest is the manner in which the transport operator B from (2.17) was split. The loss operator, L, operates on the angular flux at time $j+1$ while the scattering source operator, S, operates on the angular flux at time j. Reed notes [30] that this approach is inaccurate, particularly when transient events occur, e.g., a delta distributed source. It does have the advantage that it avoids using source iteration to solve for the within-group scattering source. The TIMEX 1972 code implements several strategies to overcome the inherent inaccuracy of this approach while still avoiding using source iteration to solve for the within-group scattering source. The details of those strategies are not relevant to my research because DI is used to solve for the within-group scattering source.

The use of diamond difference as a spatial quadrature can result in negative angular fluxes. As shown in Appendix A, this can occur when

$$\left(\frac{1}{v\Delta t} + \sigma_{\mathrm{t}}\right) >> 0, \tag{2.60}$$

hence when there are large cells, large cross sections, or short time steps. TIMEX (1972) has a zero flux fix-up scheme to account for this possibility.

One interesting feature used by TIMEX (1972) is the separation of the first flight flux and first scatter source as a mechanism to handle sources delta distributed in space and/or time. Delta distributed sources are problematic for any transport code because short time steps are required to numerically resolve the source. This presents two problems: first, the short time steps can cause the accumulation of error due to loss of precision; and second, the short time steps can result in negative fluxes.

Because the transport operator B is linear, the angular flux can be represented as $\Psi = \Psi_u + \Psi_c$ where Ψ_u is the uncollided flux and Ψ_c is the collided flux. Substituting into (2.16) yields

$$\mathbb{V}^{-1}\frac{\partial}{\partial t}\Psi_u + L\Psi_u = Q^{\mathrm{e}} \tag{2.61}$$

$$\mathbb{V}^{-1}\frac{\partial}{\partial t}\Psi_c + L\Psi_c = S(\Psi_u + \Psi_c) \tag{2.62}$$

The uncollided flux can be solved analytically and its first collision scattered neutrons, $S\Psi_u$, are then used as the fixed source when solving the collided flux numerically.

2.4.2 TIMEX (1976) Implementation. The TIMEX (1976) (Hill [16]) code is similar to the TIMEX (1972) code, the key differences being the addition of delayed neutrons and the introduction of a synthetic cross section and a synthetic source. Consider the divided difference approximation

$$\frac{\partial \psi}{\partial t} \approx \frac{\psi^{j+1} - \psi^{j}}{\Delta t} \tag{2.63}$$

where ψ^{j+1} is the angular flux at t_{j+1}, ψ^j is the angular flux at t_j, and Δt is the time interval $t_{j+1} - t_j$. Substituting (2.63) into the BTE yields

$$\frac{1}{v_g \Delta t}\left(\psi_g^{j+1} - \psi_g^{j}\right) + \hat{\Omega}\cdot\nabla\psi_g^{j+1} + \sigma_{\mathrm{t}_g}^{j+1}\psi_g^{j+1} = \sum_{g'=1}^{g-1}\sigma_{\mathrm{s}_{g'\to g}}^{j+1}\phi_{g'}^{j+1} + \sum_{g'=g}^{G}\sigma_{\mathrm{s}_{g'\to g}}^{j}\phi_{g'}^{j} + q_g^{j+1}. \tag{2.64}$$

Collecting the ψ_g^{j+1} terms and moving the ψ_g^j to the right hand side yields

$$\hat{\Omega}\cdot\nabla\psi_g^{j+1} + \left[\sigma_{\mathrm{t}_g}^{j+1} + \frac{1}{v_g\Delta t}\right]\psi_g^{j+1} = \sum_{g'=1}^{g-1}\sigma_{\mathrm{s}_{g'\to g}}^{j+1}\phi_{g'}^{j+1} + \sum_{g'=g}^{G}\sigma_{\mathrm{s}_{g'\to g}}^{j}\phi_{g'}^{j} + q_g^{j+1} + \frac{1}{v_g\Delta t}\psi_g^{j}. \tag{2.65}$$

Define the synthetic total cross section, $\tilde{\sigma}_{\mathrm{t}_g}$, as

$$\tilde{\sigma}_{\mathrm{t}_g}^{j+1} = \sigma_{\mathrm{t}_g}^{j+1} + \frac{1}{v_g \Delta t}, \tag{2.66}$$

and the synthetic source, $\tilde{q}_g$, as

$$\tilde{q}_g^{j+1} = q_g^{j+1} + \frac{1}{v_g \Delta t} \psi_g^j \tag{2.67}$$

so that

$$\hat{\Omega} \cdot \nabla \psi_g^{j+1} + \tilde{\sigma}_{\mathrm{t}_g}^{j+1} \psi_g^{j+1} = \sum_{g'=1}^{g-1} \sigma_{\mathrm{s}_{g' \to g}}^{j+1} \phi_{g'}^{j+1} + \sum_{g'=g}^{G} \sigma_{\mathrm{s}_{g' \to g}}^{j} \phi_{g'}^{j} + \tilde{q}_g^{j+1}, \tag{2.68}$$

which is in the same form as the stationary BTE. The TIMEX (1976) code used a time-independent transport code, specifically ONETRAN (Hill [15]), to solve this stationary BTE.

The authors of the TIMEX (1976) (Hill [16]) state that they would prefer to solve the implicit form

$$\mathbb{V}^{-1} \frac{\partial}{\partial t} \Psi^{j+1} + L \Psi^{j+1} = S \Psi^{j+1} + Q^{\mathrm{e}}. \tag{2.69}$$

The implicit form was not used because of the amount of computer time that would be required. For example, ONETRAN required 5.5 minutes on a CDC 7600 to solve a 20 group, 134 interval spatial mesh, S4 k_{eff} calculation (Hill [15]). The transformation of the time-dependent BTE by using the synthetic cross section points to the logical approach of using an established time-independent transport code to solve time dependent problems.

2.4.3 Implicit Backward Euler. Before proceeding to PARTISN, it is useful to examine the implicit implementation of (2.69). The implicit in time version of the within-group BTE is

$$\hat{\Omega} \cdot \nabla \psi_g^{j+1} + \tilde{\sigma}_{\mathrm{t}_g}^{j+1} \psi_g^{j+1} = \sigma_{\mathrm{s}_{g \to g}}^{j+1} \phi_g^{j+1} + \sum_{g'=1}^{g-1} \sigma_{\mathrm{s}_{g' \to g}}^{j+1} \phi_{g'}^{j+1} + \sum_{g'=g+1}^{G} \sigma_{\mathrm{s}_{g' \to g}}^{j} \phi_{g'}^{j} + \tilde{q}_g^{j+1}. \quad (2.70)$$

While both the explicit form, which is used by TIMEX, and the implicit form have first order local truncation error, the implicit form has the advantage of being numerically stable (Lewis & Miller [22]).

One of the key challenges in using the synthetic cross section is selecting the time step. Consider the monoenergetic, synthetic BTE in slab geometry (suppressing the indices for clarity)

$$\frac{\mu}{\Delta x} \left(\psi^{\mathrm{R}} - \psi^{\mathrm{L}} \right) + \tilde{\sigma}_{\mathrm{t}} \psi^{\mathrm{A}} = \sigma_{\mathrm{s}} \phi + \tilde{q}. \quad (2.71)$$

Let ϵ be the optical depth, defined as

$$\epsilon = \frac{\sigma \Delta x}{\mu}. \quad (2.72)$$

The choice of spatial quadrature will dictate a maximum optical thickness, ϵ_{tol}, that is acceptable. The maximum cell size is then

$$\Delta x_{\max} \leq \frac{|\mu_{\min}|}{\tilde{\sigma}_{\mathrm{t}}} \epsilon_{\mathrm{tol}}. \quad (2.73)$$

Clearly as $v\Delta t \to 0$, the synthetic cross section becomes large and the maximum cell size goes to zero. Conversely, as $v\Delta t \to \infty$, the time dependent behavior is lost and $\tilde{\sigma}_{\mathrm{t}} \to \sigma_{\mathrm{t}}$.

The Courant-Friedrichs-Lewy condition requires that

$$\frac{v\Delta t}{\Delta x} \leq 1. \quad (2.74)$$

Making the substitution for Δx_{max} into the Courant-Friedrichs-Lewy condition yields

$$\frac{v\Delta t}{\frac{|\mu_{\mathrm{min}}|}{\tilde{\sigma}_{\mathrm{t}}}\epsilon_{\mathrm{tol}}} \leq 1, \tag{2.75}$$

which, after substituting for $\tilde{\sigma}_{\mathrm{t}}$ to get σ_{t}, yields the constraint

$$0 < \Delta t \leq \frac{1}{\sigma_{\mathrm{t}}}\left(\frac{|\mu_{\mathrm{min}}|\epsilon_{\mathrm{tol}} - 1}{v}\right). \tag{2.76}$$

Because both σ_{t} and v are positive, then

$$|\mu_{\mathrm{min}}|\epsilon_{\mathrm{tol}} - 1 > 0 \tag{2.77}$$

must be true in order to have positive time steps. The above constraint can be written as

$$\epsilon_{\mathrm{tol}} > \frac{1}{|\mu_{\mathrm{min}}|}, \tag{2.78}$$

thus, for time dependent problems, spatial performance decreases as the angular quadrature is refined. Picking the appropriate time step requires balancing angular resolution and spatial resolution requirements. This constraint also makes the use of non-positive spatial quadratures, such as diamond-difference, in time-dependent transport problematic. Diamond-difference in slab geometry, for example, requires that

$$\Delta x < 2\frac{|\mu_{\mathrm{min}}|}{\tilde{\sigma}_{\mathrm{t}}} \tag{2.79}$$

be true to ensure positive fluxes. Applying the requirement for Δx_{max} yields the constraint

$$\epsilon_{\mathrm{tol}} < 2. \tag{2.80}$$

Thus, to maintain positivity in slab geometry, the angular quadrature must have $|\mu_{\mathrm{min}}| > 1/2$, which in the S_n angular quadrature is only true with S_2. If a negative flux fix-up scheme is implemented, the flux will be biased towards the ordinates that meet the constraint.

While it is possible for monoenergetic problems to pick a time step that results in a reasonable optical thickness, e.g.

$$\Delta t \sim \frac{1}{v\sigma_{\mathrm{t}}}, \tag{2.81}$$

the speed differential between thermal and fast neutrons makes picking a uniform time step for multigroup problems impossible. Fast neutrons are approximately 6000 times faster than thermal neutrons, which can result in synthetic cross sections that are 10 to 1000 times larger between energy groups.

Multigroup problems that only have downscatter can mitigate the time step issue by using a different time step for each energy group. However, if there is both upscatter and downscatter, this approach becomes unworkable because temporal resolution has to be created when going from a longer time step energy group to a shorter time step energy group. Similarly, if different spatial grids are used for each energy group, spatial resolution has to be created when going from a coarse grid to a fine grid.

Given the difficulty of picking an appropriate time step that works equally well for all energy groups, the best solution would be to use a spatial quadrature that is non-negative and works with optically thick cells. The alternative is to use fixups in the transport code such that it maintains adequate performance with optically thick cells. The latter approach has been taken in production codes and, thus, DI has to demonstrate equivalent performance with optically thick cells in order to be a substitute for source iteration.

2.4.4 PARTISN Implementation. PARTISN implements a semi-implicit method to discretize the time derivative in the BTE. Unlike the TIMEX codes, PARTISN is implicit in time and the extrapolated angular fluxes are second order accurate. Appendix A shows the derivation as presented by Alcouffe & Baker.

Like TIMEX (1976), PARTISN utilizes a time-independent code, in this case DANTSYS, to solve the BTE at the midpoint between time steps. If the extrapolated flux is negative, a set to zero fix-up strategy is utilized in conjunction with the balance equation. Because PARTISN uses an implicit method for solving the midpoint flux, it is numerically stable. The set to zero fix-up can result in less than second order convergence.

2.4.5 TDTORT Implementation. TDTORT implements a flux synthesis method (Goluoglu [14]), which assumes that time dependence is partially separable, i.e.,

$$\Phi(\vec{r}, \hat{\Omega}, E, t) = \Psi(\vec{r}, \hat{\Omega}, E, t)T(t). \tag{2.82}$$

The function $\Phi(\vec{r}, \hat{\Omega}, E, t)$ is assumed to be composed of a shape function, Ψ, that is slowly varying in time and an amplitude function, T, that is typically rapidly varying in time. After substituting (2.82) into the time-dependent BTE and dividing by $T(t)$, we have

$$\begin{aligned}\left[\hat{\Omega} \cdot \nabla + \sigma_{\mathrm{t}}(\vec{r}, E, t)\right] \Psi(\vec{r}, \hat{\Omega}, E, t) = &\\ \int_0^\infty \mathrm{d}E' \int \mathrm{d}\Omega' \, \sigma_s(\vec{r}, E' \to E, \hat{\Omega}' \cdot \hat{\Omega}, t)\Psi(\vec{r}, \hat{\Omega}', E', t) &\\ + q^{\mathrm{e}}(\vec{r}, \hat{\Omega}, E, t) - \frac{1}{v}\left(\frac{\mathrm{d}T(t)}{\mathrm{d}t}\frac{\Psi(\vec{r}, \hat{\Omega}, E, t)}{T(t)} + \frac{\partial}{\partial t}\Psi(\vec{r}, \hat{\Omega}, E, t)\right). &\end{aligned} \tag{2.83}$$

The advantage of decomposing the angular flux distribution in this manner is that TDTORT can utilize large time steps when solving the shape function and shorter time steps when solving the amplitude function. I did not implement this method in my code.

2.5 Time-Dependent Distribution Iteration

The key to DI is the separation of the within cell transport from the global partial currents problem. One step in affecting that separation is the integration over the spatial cell, which transforms the problem into a coupled set of partial currents that can be solved exactly. Starting from the within-group BTE in slab geometry

$$\frac{1}{v}\frac{\partial\psi_n(x,t)}{\partial t} + \mu_n\frac{\partial\psi_n(x,t)}{\partial x} + \sigma_{\mathrm{t}}(x,t)\psi_n(x,t) = q_n(x,t), \tag{2.84}$$

where n is the ordinate index, and averaging over the spatial cell yields

$$\begin{aligned}\frac{1}{\Delta x}\int_{x_L}^{x_R}\frac{1}{v}\frac{\partial\psi_n(x,t)}{\partial t}\mathrm{d}x \ + \frac{1}{\Delta x}\int_{x_L}^{x_R}\mu_n\frac{\partial\psi_n(x,t)}{\partial x}\mathrm{d}x \\ + \frac{1}{\Delta x}\int_{x_L}^{x_R}\sigma_{\mathrm{t}}(x,t)\psi_n(x,t)\mathrm{d}x \ = \frac{1}{\Delta x}\int_{x_L}^{x_R}q_n(x,t)\mathrm{d}x \ ,\end{aligned} \tag{2.85}$$

where $\Delta x = x_{\mathrm{R}} - x_{\mathrm{L}}$. Using the first mean value theorem for integration, the cross section term becomes

$$\frac{1}{\Delta x}\int_{x_L}^{x_R}\sigma_{\mathrm{t}}(x,t)\psi_n(x,t)\mathrm{d}x \ = \frac{\sigma_{\mathrm{t}}(\xi,t)}{\Delta x}\int_{x_L}^{x_R}\psi_n(x,t)\mathrm{d}x \tag{2.86}$$

where $\xi \in [x_L, x_R]$ provided that $\sigma(x,t)$ is continuous over the interval, which will be true if cells do not span material boundaries. Define the cell average angular flux and the cell average source, $\bar{q}$, as

$$\psi^{\mathrm{A}}_{n,i}(t) = \frac{1}{\Delta x}\int_{x_L}^{x_R}\psi_n(x,t)\mathrm{d}x \tag{2.87}$$

$$\bar{q}_{n,i}(t) = \frac{1}{\Delta x}\int_{x_L}^{x_R}q_n(x,t)\mathrm{d}x \ , \tag{2.88}$$

where i indexes the cell. The BTE simplifies to

$$\frac{1}{\Delta x}\int_{x_L}^{x_R}\frac{1}{v}\frac{\partial\psi_n(x,t)}{\partial t}\mathrm{d}x \ + \frac{\mu_n}{\Delta x}\left(\psi^{\mathrm{R}}_{n,i}(t) - \psi^{\mathrm{L}}_{n,i}(t)\right) + \sigma_{\mathrm{t}_i}(t)\psi^{\mathrm{A}}_{n,i}(t) = \bar{q}_{n,i}(t), \tag{2.89}$$

where $\psi^L_{n,i}$ and $\psi^R_{n,i}$ are the angular flux distributions on the left and right cell boundaries, respectively.

In order to achieve a more useful equation, the leading term needs the order of the integration and differentiation interchanged. Assuming that ψ meets the criteria for the Lebesgue dominated convergence theorem, which requires that the gradient is finite almost everywhere in the domain[2], then (because v is constant under the multigroup energy discretization)

$$\frac{1}{\Delta x}\int_{x_L}^{x_R}\frac{1}{v}\frac{\partial \psi_n(x,t)}{\partial t}\mathrm{d}x \;=\frac{1}{v\Delta x}\frac{\partial}{\partial t}\int_{x_L}^{x_R}\psi_n(x,t)\mathrm{d}x \;. \tag{2.90}$$

Thus, substituting (2.87) yields

$$\frac{1}{v}\frac{\mathrm{d}\psi^{\mathrm{A}}_{n,i}(t)}{\mathrm{d}t}+\frac{\mu_n}{\Delta x}\left(\psi^{\mathrm{R}}_{n,i}(t)-\psi^{\mathrm{L}}_{n,i}(t)\right)+\sigma_{\mathrm{t}_i}(t)\psi^{\mathrm{A}}_{n,i}(t)=\bar{q}_{n,i}(t). \tag{2.91}$$

Applying the backward Euler method to the above ordinary differential equation, yields

$$\frac{\mu_n}{\Delta x}\left(\psi^{\mathrm{R}}_{n,i,j+1}-\psi^{\mathrm{L}}_{n,i,j+1}\right)+\left(\sigma_{\mathrm{t}_{i,j+1}}+\frac{1}{v\Delta t}\right)\psi^{\mathrm{A}}_{n,i,j+1}=\bar{q}_{n,i,j+1}+\frac{1}{v\Delta t}\psi^{\mathrm{A}}_{n,i,j}, \tag{2.92}$$

where j is the time index.

A critical aspect about the synthetic total cross section approach is that it makes the transport problem more absorptive than it would be otherwise. The synthetic scattering ratio is

$$\tilde{c}=\frac{\sigma_s}{\sigma_{\mathrm{t}_{j+1}}+\frac{1}{v\Delta t}}. \tag{2.93}$$

As $v\Delta t \rightarrow 0$, the synthetic scattering ratio goes to zero. This raises the question as to whether DI will maintain a performance advantage vis-a-vis source iteration. The synthetic total cross section also makes the cells optically thick, which may impact the ability to determine the inward angular partial current distribution because of the attenuation of the angular partial currents. For short time steps, very few neu-

[2] Because this may not be strictly true for delta distributed sources, a first flux and first scatter source approach may be necessary

trons propagate across the cell, which makes the angular partial current distribution susceptible to numerical noise.

2.5.1 Derivation of Diamond Difference with Time Dependence. In this section, I will show the derivation of the diamond difference spatial quadrature transport coefficients with the time derivative present. Start with the balance equation in slab geometry, discretized with the backward Euler method

$$\frac{1}{v}\left(\frac{\psi^{\mathrm{A}}_{j+1} - \psi^{\mathrm{A}}_{j}}{\Delta t}\right) + \mu\frac{\psi^{\mathrm{R}}_{j+1} - \psi^{\mathrm{L}}_{j+1}}{\Delta x} + \sigma_{\mathrm{t}_{j+1}}\psi^{\mathrm{A}}_{j+1} = \bar{q}_{j+1}. \tag{2.94}$$

Using the auxiliary equation to eliminate ψ^{R} from the equation yields

$$\left(\sigma_{\mathrm{t}_{j+1}} + \frac{1}{v\Delta t}\right)\psi^{\mathrm{A}}_{j+1} + \frac{\mu}{\Delta x}\left(2\psi^{\mathrm{A}}_{j+1} - 2\psi^{\mathrm{L}}_{j+1}\right) = \bar{q}_{j+1} + \frac{1}{v\Delta t}\psi^{\mathrm{A}}_{j}. \tag{2.95}$$

Let $\tilde{\sigma}_{\mathrm{t}_{j+1}} = \sigma_{\mathrm{t}_{j+1}} + 1/(v\Delta t)$ and $\tilde{\epsilon}_{\mathrm{x}} = \tilde{\sigma}\Delta x/\mu$, then

$$\psi^{\mathrm{A}}_{j+1} = \frac{\bar{q}_{j+1}\frac{\Delta x}{\mu} + 2\psi^{\mathrm{L}}_{j+1} + \frac{1}{v\Delta t}\frac{\Delta x}{\mu}\psi^{\mathrm{A}}_{j}}{\tilde{\epsilon}_{\mathrm{x}} + 2}. \tag{2.96}$$

Substituting $\tilde{q}_{j+1} = \bar{q}_{j+1} + \frac{1}{v\Delta t}\psi^{\mathrm{A}}_{j}$ yields

$$\psi^{\mathrm{A}}_{j+1} = \frac{\tilde{q}_{j+1}\frac{\Delta x}{\mu} + 2\psi^{\mathrm{L}}_{j+1}}{\tilde{\epsilon}_{\mathrm{x}} + 2}, \tag{2.97}$$

which is the same form for ψ^{A} as for the time-independent case. Using the synthetic total cross section and synthetic source, any spatial quadrature can be used in a time-dependent transport code.

2.5.2 Derivation of Step Characteristic with Time Dependence. It is useful to show the derivation of a characteristic method in a time dependent form. To derive the step characteristic spatial quadrature with the time dependence term we begin with the BTE in slab geometry discretized with the backward Euler method:

$$\frac{1}{v}\left(\frac{\psi_{j+1}-\psi_j}{\Delta t}\right)+\mu\frac{\partial\psi_{j+1}}{\partial x}+\sigma_{\mathrm{t}_{j+1}}\psi_{j+1}=q_{j+1}. \tag{2.98}$$

Let $\tilde{\sigma}_{\mathrm{t}_{j+1}}=\sigma[j+1]+1/(v\Delta t)$ and dividing by μ yields

$$\frac{\partial\psi_{j+1}}{\partial x}+\frac{\tilde{\sigma}_{\mathrm{t}_{j+1}}}{\mu}\psi_{j+1}=\frac{1}{\mu}q_{j+1}+\frac{1}{\mu}\frac{\psi_j}{v\Delta t}. \tag{2.99}$$

As explained in the previous section, the cross section is constant within a cell, thus $\tilde{\sigma}_{\mathrm{t}}$ is also constant within a cell. Define $\tau(x,x_o)=(x-x_o)\tilde{\sigma}_{\mathrm{t}_{j+1}}/\mu$ where x and x_o are in the same cell. Multiplying by $e^{\tau(x,x_o)}$, we have

$$\frac{\partial\psi_{j+1}}{\partial x}e^{\tau(x,x_o)}+\frac{\tilde{\sigma}_{\mathrm{t}_{j+1}}}{\mu}\psi_{j+1}e^{\tau(x,x_o)}=\frac{1}{\mu}q_{j+1}e^{\tau(x,x_o)}+\frac{1}{\mu}\frac{\psi_j}{v\Delta t}e^{\tau(x,x_o)}. \tag{2.100}$$

Without loss of generality, we can consider the $\mu>0$ case. Noting that

$$\frac{\partial}{\partial x}\left(\psi e^{\tau(x,x_o)}\right)=\frac{\partial\psi}{\partial x}e^{\tau(x,x_o)}+\frac{\tilde{\sigma}_{\mathrm{t}}}{\mu}\psi e^{\tau(x,x_o)} \tag{2.101}$$

and integrating (2.100) from $x_o=x_{\mathrm{L}}$ to x yields

$$\begin{aligned}\psi_{j+1}(x)e^{\tau(x,x_{\mathrm{L}})}-\psi_{j+1}(x_{\mathrm{L}})e^{\tau(x_{\mathrm{L}},x_{\mathrm{L}})}=&\int_{x_{\mathrm{L}}}^{x}\mathrm{d}x'\ \frac{q_{j+1}(x')}{\mu}e^{\tau(x',x_{\mathrm{L}})}\\&+\int_{x_{\mathrm{L}}}^{x}\mathrm{d}x'\ \frac{1}{\mu v\Delta t}\psi_j(x')e^{\tau(x',x_{\mathrm{L}})}.\end{aligned} \tag{2.102}$$

Dividing by $e^{\tau(x,x_{\mathrm{L}})}$ and rearranging yields

$$\begin{aligned}\psi_{j+1}(x)=\psi_{j+1}(x_{\mathrm{L}})e^{-\tau(x,x_{\mathrm{L}})}+&\int_{x_{\mathrm{L}}}^{x}\mathrm{d}x'\ \frac{q_{j+1}(x')}{\mu}e^{\tau(x',x_{\mathrm{L}})-\tau(x,x_{\mathrm{L}})}\\&+\int_{x_{\mathrm{L}}}^{x}\mathrm{d}x'\ \frac{1}{\mu v\Delta t}\psi_j(x'))e^{\tau(x',x_{\mathrm{L}})-\tau(x,x_{\mathrm{L}})}.\end{aligned} \tag{2.103}$$

Note that

$$\tau(x',x_{\mathrm{L}})-\tau(x,x_{\mathrm{L}})=-(x-x')\tilde{\sigma}_{\mathrm{t}_{j+1}}/\mu \tag{2.104}$$

can be expressed as

$$\tau(x', x_\mathrm{L}) - \tau(x, x_\mathrm{L}) = -\tilde{\epsilon}_x (x - x')/\Delta x, \tag{2.105}$$

where $\tilde{\epsilon}_x = \tilde{\sigma}_{\mathrm{t}_{j+1}} \Delta x/\mu$. Let $x = x_\mathrm{R}$, then the transport equation for a generic cell is

$$\begin{aligned}\psi^\mathrm{R}_{j+1} = \psi^\mathrm{L}_{j+1} e^{-\tilde{\epsilon}_x} + \int_{x_\mathrm{L}}^{x_\mathrm{R}} \mathrm{d}x' \; \frac{q_{j+1}(x')}{\mu} e^{-\tilde{\epsilon}_x (x_\mathrm{R} - x')/\Delta x} \\ + \int_{x_\mathrm{L}}^{x_\mathrm{R}} \mathrm{d}x' \; \frac{1}{\mu v \Delta t} \psi_j(x') e^{-\tilde{\epsilon}_x (x_\mathrm{R} - x')/\Delta x}.\end{aligned} \tag{2.106}$$

For step characteristic, the underlying assumption is that the scattering and emission source is constant in a cell, thus

$$q_{j+1}(x) = \bar{q}_{j+1}. \tag{2.107}$$

Define the exponential moment function as (Mathews [25])

$$\mathcal{M}_n(\epsilon) = \int_0^1 (1-u)^n e^{-\epsilon u} \mathrm{d}u \; . \tag{2.108}$$

Using the transformation $u = (x_\mathrm{R} - x')/\Delta x$, the source term can be rewritten as

$$\frac{\bar{q}_{j+1}}{\mu} \int_{x_\mathrm{L}}^{x_\mathrm{R}} \mathrm{d}x' \; e^{-\tilde{\epsilon}_x (x_\mathrm{R} - x')/\Delta x} = \bar{q}_{j+1} \frac{\Delta x}{\mu} \int_0^1 \mathrm{d}u \; e^{-\tilde{\epsilon}_x u} \tag{2.109}$$

$$= \bar{q}_{j+1} \frac{\Delta x}{\mu} \mathcal{M}_0(\tilde{\epsilon}_x). \tag{2.110}$$

Note that $\bar{q}_{j+1} \Delta x/\mu$ is the integrated production rate density of particles through the cell along the ray and $\mathcal{M}_0(\tilde{\epsilon}_x)$ is the fraction that reaches the right boundary. Making the above substitution yields

$$\psi^\mathrm{R}_{j+1} = \psi^\mathrm{L}_{j+1} e^{-\tilde{\epsilon}_x} + \bar{q}_{j+1} \frac{\Delta x}{\mu} \mathcal{M}_0(\tilde{\epsilon}_x) + \int_{x_\mathrm{L}}^{x_\mathrm{R}} \mathrm{d}x' \; \frac{1}{\mu v \Delta t} \psi_j(x') e^{-\tilde{\epsilon}_x (x_\mathrm{R} - x')/\Delta x}. \tag{2.111}$$

Using the step approximation $\psi_j(x) = \psi_j^{\mathrm{A}}$ to perform the following integration

$$I = \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \mathrm{d}x' \; \psi_j(x') e^{-\tilde{\epsilon}_x (x_{\mathrm{R}} - x')/\Delta x} \tag{2.112}$$

yields

$$\frac{1}{v\Delta t} \frac{\Delta x}{\mu} \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \frac{\mathrm{d}x'}{\Delta x} \psi_j(x') e^{-\tilde{\epsilon}_x (x_{\mathrm{R}} - x')/\Delta x} = \frac{1}{v\Delta t} \psi_j^{\mathrm{A}} \frac{\Delta x}{\mu} \mathcal{M}_0(\tilde{\epsilon}_x). \tag{2.113}$$

Thus, we have

$$\psi_{i,j+1}^{\mathrm{R}} = \psi_{i,j+1}^{\mathrm{L}} e^{-\tilde{\epsilon}_x} + \left(\bar{q}_{i,j+1} + \frac{1}{v\Delta t} \psi_{i,j}^{\mathrm{A}} \right) \frac{\Delta x}{\mu} \mathcal{M}_0(\tilde{\epsilon}_x). \tag{2.114}$$

This is the same solution obtained when using the synthetic cross section and synthetic source.

The cell average flux is defined as

$$\psi_{j+1}^{\mathrm{A}} = \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \frac{\mathrm{d}x}{\Delta x} \psi_{j+1}(x). \tag{2.115}$$

Substituting in (2.103) and using the $\mu > 0$ case produces

$$\begin{aligned} \psi_{j+1}^{\mathrm{A}} = \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \frac{\mathrm{d}x}{\Delta x} \psi_{j+1}^{\mathrm{L}} e^{-\tilde{\epsilon}_x (x - x_{\mathrm{L}})/\Delta x} &+ \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \frac{\mathrm{d}x}{\Delta x} \frac{\bar{q}_{j+1}}{\mu} \int_{x_{\mathrm{L}}}^{x} \mathrm{d}x' \; e^{-\tilde{\epsilon}_x (x - x')/\Delta x} \\ &+ \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \frac{\mathrm{d}x}{\Delta x} \int_{x_{\mathrm{L}}}^{x} \mathrm{d}x' \; \frac{\psi_j(x')}{\mu v \Delta t} e^{-\tilde{\epsilon}_x (x - x')/\Delta x}. \end{aligned} \tag{2.116}$$

Using the exponential moment functions, this simplifies to

$$\begin{aligned} \psi_{j+1}^{\mathrm{A}} = \psi_{j+1}^{\mathrm{L}} \mathcal{M}_0(\tilde{\epsilon}_x) + \frac{\Delta x}{\mu} \bar{q}_{j+1} \mathcal{M}_1(\tilde{\epsilon}_x) & \\ + \int_{x_{\mathrm{L}}}^{x_{\mathrm{R}}} \frac{\mathrm{d}x}{\Delta x} \int_{x_{\mathrm{L}}}^{x} \mathrm{d}x' \; \frac{\psi_j(x')}{\mu v \Delta t} e^{-\tilde{\epsilon}_x (x - x')/\Delta x}&. \end{aligned} \tag{2.117}$$

Next, the following integration needs to be performed

$$I = \int_{x_\mathrm{L}}^{x_\mathrm{R}} \frac{\mathrm{d}x}{\Delta x} \int_{x_\mathrm{L}}^{x} \mathrm{d}x' \; \frac{\psi_j(x')}{\mu v \Delta t} e^{-\tilde{\epsilon}_x (x - x')/\Delta x}. \tag{2.118}$$

Using the step approximation to solve the integral I yields

$$\psi_{j+1}^{\mathrm{A}} = \psi_{j+1}^{\mathrm{L}} \mathcal{M}_0(\tilde{\epsilon}_x) + \frac{\Delta x}{\mu} \left(\bar{q}_{j+1} + \frac{\psi_j^{\mathrm{A}}}{v \Delta t} \right) \mathcal{M}_1(\tilde{\epsilon}_x). \tag{2.119}$$

This is, again, the same equation obtained by using the synthetic cross section and synthetic source.

III. Matrix Albedo

While developing the verification plan (Chapter V) there was a need for a non-discrete ordinates benchmark that was more accurate than a Monte Carlo code. Ganapol [13] developed the TIEL benchmark, which uses a Fourier transform solution based on an analytical moments representation of the Green's function. The benchmark features a delta-distributed source in an infinite medium. The TIEL benchmark is a semi-analytic solution to

$$\left[\mu\frac{\partial}{\partial x}+1\right]\psi(x,\mu)=\frac{c}{2}\sum_{l=0}^{L}\omega_l P_l(\mu)\psi_l(x)+\frac{\delta(x)}{2} \tag{3.1}$$

where ω_l are the scattering moments, P_l are the Legendre polynomials and $\psi_l(x)$ are the Legendre moments given by

$$\psi_l(x)=\frac{1}{2}\int_{-1}^{1}\mathrm{d}\mu\ P_l(\mu)\psi(x,\mu). \tag{3.2}$$

This benchmark presents two challenging problems. First, the delta-distributed source must be adequately represented in the transport code and, second, the infinite medium has to be properly handled. The other motivation for implementing this benchmark is that it is similar to the monoenergetic, time dependent benchmark also developed by Ganapol [12].

The delta-distributed source can be implemented in two different fashions. The first method would be to introduce a single cell source region; however, this is a crude approach because it effectively distributes the source into a small volume. An alternative approach would be to use a boundary current in conjunction with a symmetry boundary condition.

Usually, an incident current is specified as a Lambertian source, i.e., the incident flux is independent of direction. Alternatively, one can specify a source such that the incident *current* is isotropically distributed (Appendix C). The delta-distributed source in the TIEL benchmark is equivalent to an isotropically distributed incident current. Isotropic currents are particularly challenging because the incident flux is singular as $\mu \to 0$. The accuracy of the incident current representation, therefore, is dependent upon the order of the angular quadrature.

The implementation of an incident current with a non-vacuum boundary condition is not a common feature found in production transport codes. I did implement it in my DI code because it provided two different methods for utilizing the TIEL benchmark.

3.1 Infinite Medium

While the treatment of the delta-distributed source is relatively straightforward, the infinite medium is not as easy. There are traditionally two different approaches to handling an infinite medium. The first method is to define a region much larger than the region of interest and use vacuum boundaries. For example, if the region of interest is 4 mean free paths from the origin, the problem might be defined as a 100 mean free path region. The second method is to utilize either a specular or grey boundary condition and define a region smaller than the first method. Both of these methods are deficient when the scattering ratio is non-zero. The first method will underestimate the scalar flux because the contribution from the medium beyond the defined region is omitted. The reflected angular flux distribution produced by either the specular or the grey boundary conditions will be inaccurate (Figure 3.1).

None of these methods were acceptable in the verification testing because of the difficulty of attributing the cause of any discrepancies between the results produced by DI and the TIEL solution. Instead, a boundary condition that produces an accurate reflected flux is required to reliably conduct the verification testing. Such a boundary condition could be implemented as a matrix, which properly distributes the outgoing flux to match the returning flux from an infinite medium.

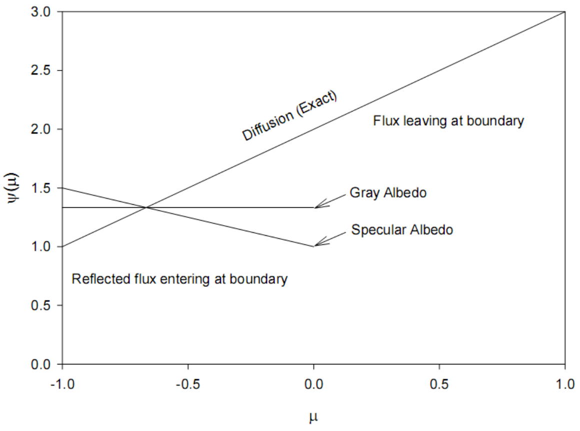


Figure 3.1: Conceptual flux distributions at a boundary.

3.2 Matrix Albedo in Slab Geometry

Consider a region within the infinite medium that encompasses any points of interest. Without loss of generality, consider only the right boundary by adding one cell to the right boundary of the defined region. Define the angular flux vectors entering and exiting the additional cell as shown in Figure 3.2. The outgoing fluxes

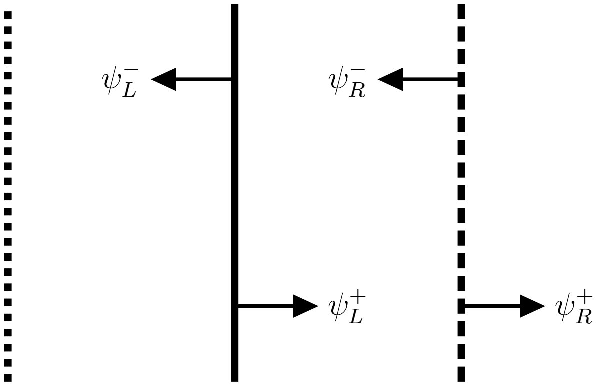


Figure 3.2: Region of interest with the one cell extension.

(ψ_L^- and ψ_R^+) are related to the incoming fluxes (ψ_L^+ and ψ_R^-) through the expression

$$\begin{bmatrix} \psi_L^- \\ \psi_R^+ \end{bmatrix} = \begin{bmatrix} \mathbb{M}_{\text{LR}} & \mathbb{M}_{\text{LL}} \\ \mathbb{M}_{\text{RR}} & \mathbb{M}_{\text{RL}} \end{bmatrix} \begin{bmatrix} \psi_R^- \\ \psi_L^+ \end{bmatrix}, \tag{3.3}$$

where $\mathbb{M}_{\mathrm{LR}}$ is the transmission coefficient matrix from the right face to the left face, $\mathbb{M}_{\mathrm{LL}}$ is the reflection coefficient matrix for the left face, $\mathbb{M}_{\mathrm{RR}}$ is the reflection coefficient matrix for the right face and $\mathbb{M}_{\mathrm{RL}}$ is the transmission coefficient matrix from the left face to the right face. Recall that (2.35) expresses the relationship between the outgoing currents (or fluxes) and incoming currents (or fluxes). Thus, the coefficient matrices are extracted from the $\mathbf{m}^{\mathrm{oi}}$ matrix on a flow direction basis. The columns of the $\mathbb{M}$ matrices correspond to the inward flow and the rows correspond to the outward flow. The $\mathbb{M}_{\mathrm{LL}}$ matrix, for example, is constructed by populating the elements where the columns map to the positive ordinates and the rows map to the negative ordinates.

Define a matrix albedo at the boundar of the region of interest, $\mathbb{A}$, such that

$$\psi_L^- = \mathbb{A}\psi_L^+. \tag{3.4}$$

Because the extra cell is one cell of an infinite medium, the medium to the right of the *right* boundary of the cell is identical to the medium to the right of the *left* boundary of the cell, thus

$$\psi_R^- = \mathbb{A}\psi_R^+. \tag{3.5}$$

If the incident flux ψ_L^+ is replaced with the identity matrix, which effectively is a matrix constructed from the unit incident flux vectors for each ordinate, then

$$\Psi_L^- = \mathbb{A}\mathbb{I} = \mathbb{A}, \tag{3.6}$$

where Ψ_L^- is the return flux matrix. Let $\mathbb{F}$ be the incident flux matrix on the right boundary, then

$$\Psi_R^- = \mathbb{A}\mathbb{F}. \tag{3.7}$$

Substituting into (3.3) yields

$$\begin{bmatrix} \mathbb{A} \\ \mathbb{F} \end{bmatrix} = \begin{bmatrix} \mathbb{M}_{\mathrm{LR}} & \mathbb{M}_{\mathrm{LL}} \\ \mathbb{M}_{\mathrm{RR}} & \mathbb{M}_{\mathrm{RL}} \end{bmatrix} \begin{bmatrix} \mathbb{A}\mathbb{F} \\ \mathbb{I} \end{bmatrix}. \tag{3.8}$$

One obvious method for solving the above equation for $\mathbb{A}$ is a form of source iteration. Consider a flight of neutrons that enter the cell from the left. Some are reflected back ($\mathbb{A}_0 = \mathbb{M}_{\mathrm{LL}}\mathbb{I}$) and some propagate to the right boundary ($\mathbb{F}_0 = \mathbb{M}_{\mathrm{RL}}\mathbb{I}$). A second flight of neutrons enter the cell from the left. Again, some neutrons are reflected plus there is a contribution from the neutrons that reflected after exiting the right boundary ($\mathbb{A}_1 = \mathbb{M}_{\mathrm{LL}}\mathbb{I} + \mathbb{M}_{\mathrm{LR}}\mathbb{A}_0\mathbb{F}_0$). The algorithm that implements this iteration is shown in Algorithm 3.1. An alternative iteration strategy is shown in Algorithm 3.2. I implemented both methods and there was no difference in performance.

```
Compute $\mathbb{U} = \mathbb{M}_{\mathrm{LL}}\mathbb{M}_{\mathrm{RL}}$
repeat
  Compute $\mathbb{A} = \mathbb{M}_{\mathrm{LR}}\mathbb{U} + \mathbb{M}_{\mathrm{LL}}$
  Compute $\mathbb{F} = \mathbb{M}_{\mathrm{RR}}\mathbb{U} + \mathbb{M}_{\mathrm{RL}}$
  Compute $\mathbb{U} = \mathbb{A}\mathbb{F}$
until $\mathbb{U}$ converges
```

Algorithm 3.1: Iteration algorithm for determining the matrix albedo

```
Compute $\mathbb{U} = \mathbb{M}_{\mathrm{LL}}\mathbb{M}_{\mathrm{RL}}$
repeat
  Compute $\mathbb{A} = \mathbb{M}_{\mathrm{LR}}\mathbb{U} + \mathbb{M}_{\mathrm{LL}}$
  Solve $(\mathbb{I} - \mathbb{A}\mathbb{M}_{\mathrm{RR}})\mathbb{U} = \mathbb{A}\mathbb{M}_{\mathrm{RL}}$ for $\mathbb{U}$
until $\mathbb{U}$ converges
```

Algorithm 3.2: Alternative iteration algorithm for determining the matrix albedo

IV. Implementation

I pursued my research effort in two parallel tracks. The first track was to develop a robust time independent distribution iteration code. The second track was to adapt the distribution iteration method into a time dependent scheme. The essential element of the first track was the application of modern software engineering methods to a nuclear engineering problem. The second track required the assessment of the various mathematical treatments of the discretization of the time derivative. These established the requirements for my implementation efforts.

4.1 Development of Time Independent Distribution Iteration Code

The implementation of the time-independent one-dimensional slab and two-dimensional XY geometries followed the work of Wager [32] and Prins [28] closely. Prins presented a block vector and matrix notation that captured the matrices and vectors associated with X and Y dimensions into one set of equations. I followed a logical approach and extended the notation to include the vectors and matrices associated with the Z dimension.

4.1.1 Software Engineering Practices. I quickly decided to start with a blank sheet and not reuse the software developed by Wager and Prins. The primary reason was that during the course of their work, many avenues were explored which contributed to needless complexity and residual constraints of their implementations of the DI method. Cleaning the existing code would have been harder than starting over. Another reason for starting over was the goal to extend DI into three dimensional XYZ geometry. Having XYZ geometry as a design goal impacts the design of data structures. The third reason was the fact that a time dependent algorithm was going to be implemented using DI.

In the development of my DI code, I adopted several key practices that are used in the software engineering field, including:

1. Using a source code management system;

2. Testing to ensure that revisions do not prevent compilation or introduce error;
3. Committing changes to the source management system frequently.

The two key features that I wanted from a source code management system were the ability to branch and to rollback changes. The ability to branch, that is to have a separate development effort, from the main development effort is an important feature to have in order to support concurrent research efforts. The second and third practices resulted in a coding work flow that focused on making small incremental changes and frequent testing.

Based on the development practices that I adopted, the strategy for developing the DI code was to first implement a slab geometry version and verify its performance. Once a working implementation of slab geometry was available, even though the feature set was incomplete, the functional slab geometry was branched from the main development effort. The code was then reviewed for the elements common between XY and slab geometry. The common elements were separated from the geometry specific elements and two libraries were created. An XY geometry specific library was created, using the slab geometry specific library as a template. This resulted in a functional code that consisted of a verified slab geometry component and an unverified XY geometry component.

The main trunk development at this point was focused on verifying the performance of the XY geometry and correcting errors. The test suite consisted of both slab and XY geometry cases and changes were reviewed to make sure slab geometry performance was not affected. Additional features were added to slab geometry to support research efforts outside the scope of this dissertation. Support for time dependent transport was added and testing in slab geometry was performed. The introduction of time dependent support at this juncture was done to ensure that the evolution of the source code would be compatible with the goals of my research.

Once a functional and verified XY geometry implementation was available, the slab and XY geometry implementation was branched and development of a XYZ geometry implementation was started in the main trunk. The XY geometry specific library was used as a template for the XYZ geometry. During the course of the development of the XYZ geometry version, additional code elements that would be common to all three geometries became evident; however, I did not attempt to extract these newly identified common elements because they required some substantial changes to the code base. In keeping with the development principles that I adopted, the extraction of the common elements would require a parallel development effort and then a subsequent merge of the changes into the main development trunk. The extraction of these common elements was deferred at this time; however, I recommend that it be pursued prior to any work in developing a message passing interface parallelized version.

4.2 Three Dimensional Geometry

The first step in developing the three-dimensional implementation of DI, or any transport code for that matter, is to establish the cell orientation. Because I wanted to maintain the orientation where the XY plane is normal to an observer's line of sight, the Z axis extends towards the observer in the positive Z direction. This arrangement forms the basis of the arrangement of the cell faces, as shown in Figure 4.1. While this arrangement is purely arbitrary, it does facilitate code reuse with the XY and slab geometries.

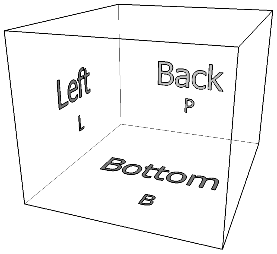


Figure 4.1: Three-Dimensional Cube Orientation and Face Naming

Given the arrangement of faces of a computational cell, the next step is to establish an ordering convention for use when constructing the partial currents linear system. Again, I kept the order established in the XY geometry and added the Z faces after the XY faces. This results in an arrangement where all the X faces (YZ planes) are first, then the Y faces (XZ planes) and finally the Z faces (XY planes). When implementing DI, one has to decide on whether to use angular fluxes or angular currents to represent flows into and out of the cell. I used the current representation because Prins [28] found that DI converges more rapidly with it than with the angular flux representation.

4.2.1 Zeroth Spatial Moment Methods. In XYZ geometry, I adopted the block vector and matrix notation used by Prins [28]. The cell face angular current vector is

$$\vec{j} = \begin{pmatrix} \vec{j}_L & \vec{j}_R & \vec{j}_B & \vec{j}_T & \vec{j}_P & \vec{j}_F \end{pmatrix}^T , \tag{4.1}$$

where $\vec{j}_L$ is the current on the left face, $\vec{j}_R$ is the current on the right face, $\vec{j}_B$ is the current on the bottom face, $\vec{j}_T$ is the current on the top face, $\vec{j}_P$ is the current on the back face[1] and $\vec{j}_F$ is the current on the front face. The scattering and emission source in a cell for a zeroth moment method is simply

$$q^{\mathrm{s}} = \bar{q}^{\mathrm{s}} \tag{4.2}$$

and

$$q^{\mathrm{e}} = \bar{q}^{\mathrm{e}} \tag{4.3}$$

where $\bar{q}^{\mathrm{s}}$ is the cell average scattering source and $\bar{q}^{\mathrm{e}}$ is the cell average emission source. The system of discrete ordinates spatial quadrature equations for within-cell transport is

$$\vec{j}^{\,\mathrm{out}} = \mathbb{K}^{\mathrm{OI}} \vec{j}^{\,\mathrm{in}} + \mathbb{K}^{\mathrm{OS}} q^{\mathrm{s}} + \mathbb{K}^{\mathrm{OE}} q^{\mathrm{e}} \tag{4.4}$$

and

$$\psi^{\mathrm{A}} = \mathbb{K}^{\psi \mathrm{I}} \vec{j}^{\,\mathrm{in}} + \mathbb{K}^{\psi \mathrm{S}} q^{\mathrm{s}} + \mathbb{K}^{\psi \mathrm{E}} q^{\mathrm{e}}. \tag{4.5}$$

[1] The superscript "P" is from "Posterior" or, if you prefer, the Greek translation of back is $\pi\iota\sigma\omega$.

As in the XY geometry case, we define the augmented $\mathbb{K}^{\mathrm{OI}}$ matrix:

$$\mathbb{K}^{\mathrm{OI}} = \begin{pmatrix} \mathbb{K}^{\mathrm{OI}}_{LL} & \mathbb{K}^{\mathrm{OI}}_{LR} & \mathbb{K}^{\mathrm{OI}}_{LB} & \mathbb{K}^{\mathrm{OI}}_{LT} & \mathbb{K}^{\mathrm{OI}}_{LP} & \mathbb{K}^{\mathrm{OI}}_{LF} \\ \mathbb{K}^{\mathrm{OI}}_{RL} & \mathbb{K}^{\mathrm{OI}}_{RR} & \mathbb{K}^{\mathrm{OI}}_{RB} & \mathbb{K}^{\mathrm{OI}}_{RT} & \mathbb{K}^{\mathrm{OI}}_{RP} & \mathbb{K}^{\mathrm{OI}}_{RF} \\ \mathbb{K}^{\mathrm{OI}}_{BL} & \mathbb{K}^{\mathrm{OI}}_{BR} & \mathbb{K}^{\mathrm{OI}}_{BB} & \mathbb{K}^{\mathrm{OI}}_{BT} & \mathbb{K}^{\mathrm{OI}}_{BP} & \mathbb{K}^{\mathrm{OI}}_{BF} \\ \mathbb{K}^{\mathrm{OI}}_{TL} & \mathbb{K}^{\mathrm{OI}}_{TR} & \mathbb{K}^{\mathrm{OI}}_{TB} & \mathbb{K}^{\mathrm{OI}}_{TT} & \mathbb{K}^{\mathrm{OI}}_{TP} & \mathbb{K}^{\mathrm{OI}}_{TF} \\ \mathbb{K}^{\mathrm{OI}}_{PL} & \mathbb{K}^{\mathrm{OI}}_{PR} & \mathbb{K}^{\mathrm{OI}}_{PB} & \mathbb{K}^{\mathrm{OI}}_{PT} & \mathbb{K}^{\mathrm{OI}}_{PP} & \mathbb{K}^{\mathrm{OI}}_{PF} \\ \mathbb{K}^{\mathrm{OI}}_{FL} & \mathbb{K}^{\mathrm{OI}}_{FR} & \mathbb{K}^{\mathrm{OI}}_{FB} & \mathbb{K}^{\mathrm{OI}}_{FT} & \mathbb{K}^{\mathrm{OI}}_{FP} & \mathbb{K}^{\mathrm{OI}}_{FF} \end{pmatrix}. \tag{4.6}$$

The subscripts denote the combination of cell faces using the notation (*to*, *from*), e.g., the subscript “LT” is for flows to the Left face from the Top face. The submatrices, e.g. $\mathbb{K}^{\mathrm{OI}}_{LL}$, operate on a subvector of angular partial currents and are sized by the number of ordinates. The $\mathbb{K}^{\mathrm{OI}}$ matrix has a very sparse structure because there is no coupling between ordinates as a result of scattering. All the submatrices along the main diagonal, e.g. $\mathbb{K}^{\mathrm{OI}}_{LL}$, are zero matrices. The off-diagonal submatrices, e.g. $\mathbb{K}^{\mathrm{OI}}_{LR}$, only have non-zero elements where there is a coupling between an ordinate on the inflow face and an ordinate on the outflow face. The location of the non-zero elements, as well as the total number, in each submatrix depends upon the angular quadrature used. For example, in the S_n angular quadrature there are $n(n+2)$ ordinates, which results in a $\mathbb{K}^{\mathrm{OI}}$ matrix with dimensions of $6n(n+2) \times 6n(n+2)$. Each off-diagonal submatrix has $n(n+2)/2$ non-zero elements along its main diagonal for a total of $15n(n+2)$.

The $\mathbb{K}^{\mathrm{OS}}$ and $\mathbb{K}^{\mathrm{OE}}$ matrices are

$$\mathbb{K}^{\mathrm{OS}} = \begin{pmatrix} \mathbb{K}^{\mathrm{OS}}_{L} & \mathbb{K}^{\mathrm{OS}}_{R} & \mathbb{K}^{\mathrm{OS}}_{B} & \mathbb{K}^{\mathrm{OS}}_{T} & \mathbb{K}^{\mathrm{OS}}_{P} & \mathbb{K}^{\mathrm{OS}}_{F} \end{pmatrix}^{T} \tag{4.7}$$

$$\mathbb{K}^{\mathrm{OE}} = \begin{pmatrix} \mathbb{K}^{\mathrm{OE}}_{L} & \mathbb{K}^{\mathrm{OE}}_{R} & \mathbb{K}^{\mathrm{OE}}_{B} & \mathbb{K}^{\mathrm{OE}}_{T} & \mathbb{K}^{\mathrm{OE}}_{P} & \mathbb{K}^{\mathrm{OE}}_{F} \end{pmatrix}^{T}. \tag{4.8}$$

For the above matrices, the subscript denotes the outflow face. Both the $\mathbb{K}^{\mathrm{OS}}$ and $\mathbb{K}^{\mathrm{OE}}$ matrices are sparse–the non-zero elements are where an ordinate is in the outward direction for a face. For example, the $\mathbb{K}_L^{\mathrm{OE}}$ submatrix has non-zero elements for ordinates where $\mu < 0$.

The $\mathbb{K}^{\psi\mathrm{I}}$ matrix is defined as

$$\mathbb{K}^{\psi\mathrm{I}} = \begin{pmatrix} \mathbb{K}_L^{\psi\mathrm{I}} & \mathbb{K}_R^{\psi\mathrm{I}} & \mathbb{K}_B^{\psi\mathrm{I}} & \mathbb{K}_T^{\psi\mathrm{I}} & \mathbb{K}_P^{\psi\mathrm{I}} & \mathbb{K}_F^{\psi\mathrm{I}} \end{pmatrix}. \tag{4.9}$$

In the case of the $\mathbb{K}^{\psi\mathrm{I}}$ matrix, the subscript denotes the inflow face. The $\mathbb{K}^{\psi\mathrm{I}}$ matrix is sparse as each submatrix is diagonal. For zeroth moment methods, the $\mathbb{K}^{\psi\mathrm{S}}$ and $\mathbb{K}^{\psi\mathrm{E}}$ matrices are the same in slab, XY and XYZ geometries. The augmented scattering matrix is defined as

$$\Sigma = \begin{pmatrix} \Sigma_{\mathrm{s}} & 0 & 0 \\ 0 & \Sigma_{\mathrm{s}} & 0 \\ 0 & 0 & \Sigma_{\mathrm{s}} \end{pmatrix}, \tag{4.10}$$

where Σ_{s} is the isotropic scattering matrix. For anisotropic scatter each submatrix will be different. The Σ_{s} matrix is not sparse, though the augmented matrix, Σ, is sparse.

While the augmented $\mathbb{K}$ and Σ matrices are large, the primary benefit of using them is that the implementation becomes simpler. The large dimensions of the matrices can be mitigated by representing them as sparse matrices and using a sparse Basic Linear Algebra System library. For example, in the S_8 angular quadrature, the $\mathbb{K}^{\mathrm{OI}}$ matrix is 480 × 480 with 2000 non-zero elements, which is less than one percent of all the elements. The $\mathbb{K}^{\mathrm{OE}}$ and $\mathbb{K}^{\mathrm{OS}}$ are 480 × 80 with 240 non-zero elements and the $\mathbb{K}^{\psi\mathrm{I}}$ matrix is 80 × 480 with 480 non-zero elements. The Σ matrix is the most dense matrix with a third of its elements being non-zero.

The solution of the within-cell transport equations follows directly from Mathews [24]. By using the augmented vector and matrix notation, we have the same set of equations in slab, XY, and XYZ geometry.

4.2.2 First Spatial Moment Methods. All the vectors and matrices can be further expanded by adding the components for the first spatial moments in a block fashion as demonstrated by Prins [24]. The use of the block notation allows the implementation of a unified code that handles both zero and first spatial moment methods.

4.2.3 Boundary Conditions. Boundary conditions can either be explicit or implicit. An explicit boundary condition provides a known incoming flux (or current) on the surface of the problem domain. An example of a explicit boundary condition is a Lambertian illumination (Appendix C). An implicit boundary condition redistributes the outward flux back into the problem domain.

An incident illumination can either be specified as an angular flux, which is the typical approach, or as a current. My DI code supports both notations to facilitate my verification effort. Because I used the current representation, angular fluxes need to be converted into partial currents (both $\vec{J}$ and $\vec{j}$). For simplicity and to facilitate angular quadrature refinement testing, I only implemented an isotropic angular flux (a Lambertian illumination)—an anisotropic angular flux is a trivial modification of the input file processor. For an incident current, the partial current and a distribution (Lambertian or Isotropic Surface Source) is specified as input. The code then computes the angular partial current using the specified distribution.

Given an incident angular flux $\psi(\hat{\Omega})$, the inward partial current on face, J_n^{in} is given by

$$J_n^{\mathrm{in}} = \int_{\hat{n}\cdot\hat{\Omega}<0} \mathrm{d}\Omega \; |\hat{n}\cdot\hat{\Omega}|\psi(\hat{\Omega}) \tag{4.11}$$

where the subscript n denotes the face and $\hat{n}$ is the outward unit vector normal to the face. The inward current on a cell is $\vec{j}_n^{\mathrm{in}}(\hat{\Omega}) = |\hat{n}\cdot\hat{\Omega}|\psi(\hat{\Omega})$.

4.3 Implementing the Matrix Albedo Into Distribution Iteration

The matrix albedo was implemented in my DI code for slab geometry. Several changes are required to implement the matrix albedo:

1. Calculate the transport coefficients, e.g. $\mathbb{K}^{\mathrm{OI}}$, for the infinite medium material.
2. Calculate the $\mathbf{m}^{\mathrm{oi}}$ matrix for the infinite medium material.
3. Implement a solver for the albedo matrix A.
4. Implement all boundary conditions, e.g. specular reflection, in matrix form and use them when improving the inflow flux distribution.
5. Provide a routine to compute the reflection coefficient used in the partial current solver.

Steps 1 and 2 were easy to implement in my DI code because the code necessary to compute the coefficient matrices and $\mathbf{m}^{\mathrm{oi}}$ was already implemented. The code was implemented to permit the use of different materials on the left and right boundaries. For example, a three region problem could have an infinite region of material "A" on the left, a region of interest of finite length of material "B" and an infinite region of material "C" on the left.

I implemented two different solvers for the matrix albedo (Mathews & Dishaw [23]). Because both solvers are iterative solvers, care must be taken to avoid false convergence. Both algorithms check for convergence using an iteration count doubling strategy shown in Algorithm 4.1. This algorithm produced reliable results even in a strongly scattering medium, e.g. $c = 0.9$.

In order to facilitate the use of specular and grey boundaries along with the infinite medium matrix albedo boundary, all boundaries were represented with a matrix. A specular boundary can be represented in matrix form as (where ordinate 1 and n

```
Set iteration count to 1
Set check convergence to 2
Initialize $\mathbb{U}_\mathrm{p}$ to zero
repeat
  Solve for $\mathbb{U}$, e.g. Algorithm 3.1
  if iteration count is equal to check convergence then
    if Symmetric relative difference of $\mathbb{U}$ and $\mathbb{U}_\mathrm{p}$ less than tolerance then
      Exit the iteration
    end if
    $\mathbb{U}_\mathrm{p} = \mathbb{U}$
    Double the value of check convergence
  end if
  Increment iteration count
until Maximum number of iterations are exceeded
if Maximum number of iterations were exceeded then
  Signal failure to converge
end if
```

Algorithm 4.1: Iteration count algorithm used to determine if the matrix albedo has converged.

form a reflection pair, 2 and $n-1$ form another reflection pair, etc.)

$$\begin{bmatrix} j_{n/2+1} \\ j_{n/2+2} \\ \vdots \\ j_{n-1} \\ j_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & \ldots & 0 & \alpha \\ 0 & 0 & \ldots & \alpha & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \alpha & \ldots & 0 & 0 \\ \alpha & 0 & \ldots & 0 & 0 \end{bmatrix} \begin{bmatrix} j_1 \\ j_2 \\ \vdots \\ j_{n/2-1} \\ j_{n/2} \end{bmatrix}, \tag{4.12}$$

where α is the reflection coefficient ($\alpha = 1$ is a symmetry boundary, $\alpha < 1$ is a dirty mirror). A grey boundary would have each row of the matrix filled in.

The matrix albedo requires the implementation of a method to determine the reflection coefficient that will be used in the partial current solver. The reflection coefficient at a face is defined as

$$\alpha_{face} = \frac{J^{\mathrm{in}}_{face}}{J^{\mathrm{out}}_{face}}, \tag{4.13}$$

where J^{out} is the partial current leaving the region and J^{in} is the partial current that has been reflected back. Let $\mathbb{A}_{face}$ be the matrix albedo then, by definition,

$$J^{\text{in}} = \vec{w} \cdot \mathbb{A}_{face} \mathbb{Z}_{face} J^{\text{out}}, \tag{4.14}$$

where $\vec{w}$ is a vector of ordinate weights and $\mathbb{Z}_{face}$ is the angular current distribution for a face, which is a vector. The reflection coefficient is, therefore,

$$\alpha = \vec{w} \cdot \mathbb{A}_{face} \mathbb{Z}_{face}. \tag{4.15}$$

4.4 Multigroup and Time Dependent Transport

The conventional approach for implementing time dependent and multigroup transport is through an outer-iteration method. For example, in a time-independent multigroup code the outer-iteration starts at the highest energy group and uses a time-independent monoenergetic transport solver to determine the angular (or scalar) fluxes for each energy group using the other group angular fluxes as part of the external source for the within-group problem. For a problem where neutrons only scatter downwards in energy, only one outer-iteration is required because the lower energy groups do not couple to the higher energy groups. A problem that includes the upscatter of neutrons or fission neutrons, multiple outer-iterations are required because of the coupling of the energy groups.

Using the synthetic total cross-section and synthetic source method, time-dependent problems can be solved using an outer-iteration method. Unlike multigroup transport, only one iteration for all the time steps is required because there is no scattering of neutrons to previous time steps. A combined time-dependent, multigroup transport code would then consist of two outer-iterations, with the time-dependent outer-iteration calling the multigroup iteration.

The implementation of a multigroup transport code involves a mechanism for storing the group angular flux at each time step, an iteration that solves each energy group until all the group angular fluxes are converged, and a subroutine for calculating the within-group external source. When using the DI method, storing the inward angular partial current distribution for each energy group as the initialization (vice isotropic) will offer a performance advantage.

The implicit backward Euler method serves as a useful starting point for adapting the DI method into a time-dependent algorithm. As shown in Section 2.5, the DI method can be derived with the time-dependent form of the Boltzmann Transport Equation. This method does not require any additional storage if careful use of the cell average angular flux data structure is observed. The midpoint average method that PARTISN utilizes does require a second copy of the cell average angular flux data structure, thus the price in memory has to be weighed against the improved accuracy in time. As in the multigroup implementation, a mechanism for storing and retrieving the inward angular partial current distribution for the previous time step will offer a performance advantage.

V. Verification

To support my research, a formal approach to verification of the DI method was needed. The primary reason for pursuing formal verification was to provide a measure of confidence in the underlying code base before implementing the time dependent version of DI. The secondary reason was to build upon the work performed by Wager [32] and Prins [28] in establishing confidence in the DI method as a viable alternative to the more established source-iteration based methods.

In the vernacular of the software testing field, the formalized testing of software is described as "validation and verification" (V&V). Validation is a measure of the software's ability to perform in its intended application, e.g. the model is an adequate analog to reality for the intended application. Verification is a measurement of the accuracy and reliability of the implementation of a model. Verification does not ascertain whether or not the implementation is suitable for an application. The fidelity of the model combined with the verification of its implementation provides an expectation of whether the combination is valid for a specified application.

For the purpose of my research, I focused on verification and did not pursue validation. Because the discrete ordinates method which underlies DI has been studied and analyzed for forty years and its applicability is well understood in the nuclear engineering community. Furthermore, the focus of my research was to develop a more robust implementation of the discrete ordinates method rather than to design towards a specific application of discrete ordinates.

There is a hierarchy of models that is intimately part of a V&V effort. First, there is the physical model which serves as the description of some physical process. Next is the mathematical model that expresses the physical model. The mathematical model can be expressed in a form to facilitate its numerical solution, e.g., discretization. Finally there is the computer model, which is the model that is ultimately implemented in software. Any one of these models can be the subject of validation or verification.

The purpose of verification is to identify deviations between the output of the implementation of a model and the expected values [31]. The source of these deviations can be attributed to uncertainty and error. Uncertainty can be further categorized as reducible (epistemic) and irreducible (aleatoric). Epistemic uncertainty is the result of insufficient information about the physical system. Aleatoric uncertainty is due to the probabilistic distribution of the inputs, e.g. variation in the material properties. Because the goal of my research is to develop an improved algorithm for solving discrete ordinates, the effects of epistemic and aleatoric uncertainty are irrelevant and need to be eliminated. Specifically, my research is focused on verifying that DI does not introduce error in the solution of the discrete ordinates equations for a given spatial and angular quadrature. In the aforementioned hierarchy of models, my research is comparing the computer model, DI, to the numerical model. Thus, all sources of uncertainty must be eliminated; for example, aleatoric uncertainty is eliminated through the use of specified material properties rather than the use of real material properties. This permits me to identify any sources of error in my implementation rather than attempting to discriminate between deviations due to uncertainty and those due to error. Because DI is based on discrete ordinates—thereby inheriting its benefits and deficiencies—the use of a discrete ordinates based benchmark is not a significant shortcoming. If experimental data were used instead, the effects of epistemic and aleatoric uncertainty would need to be identified.

The verification approach that I implemented is a comparison with another implementation of the discrete ordinates equations, specifically PARTISN version 4.00 (beta release 05-26-04)[1] (Alcouffe [2]). A suite of test problems was constructed and a suitable benchmark was selected. While the preferred benchmark is an analytic solution, there is, unfortunately, a paucity of analytic benchmarks for the Boltzmann Transport Equation. I categorize the benchmarks into the following classes:

- Analytic solutions,

[1] As of 21 July 2007 this was the only version available for distribution. Los Alamos National Laboratory has not made a newer version available

- Semi-analytic solution (e.g. Ganapol's benchmarks [12] and [13]),
- Exact discretized solutions,
- Problems constructed to have specified solutions, i.e., manufactured problems, and
- Established codes (e.g. PARTISN).

The use of analytic and semi-analytic benchmarks is not necessarily required for the type of verification that I performed for my research. The analytic and semi-analytic benchmarks are for verification of the underlying numerical model–discrete ordinates–to the mathematical model–the Boltzmann Transport Equation. I included them because I wanted to confirm that I was using PARTISN correctly and was not introducing a common error. I did not use any manufactured problems in the verification of my DI code.

5.1 Time-Independent Verification Tests

The suite of tests is first divided into the three supported geometries (Slab, 2D rectangular, and 3D boxoid). Grouping by geometry is required as the different types of geometries introduce different testing requirements. The tests are further subdivided as to whether the intent is to test a qualitative or quantitative aspect. This subdivision is more for software development convenience than any other reason. The qualitative tests were designed with automation in mind so that they could be run frequently, typically before the source code was checked into the revision management system and also during automated testing of a neutral build. This allowed for a quick determination of whether the software development effort was on track. The quantitative tests were designed to assess the correctness of the DI code. These tests required greater analytical effort and were not fully automated.

Wager [32] and Prins [28] both demonstrated the performance of the Distribution Iteration method for a series of time-independent problems. Wager developed DI originally and examined its performance in slab geometry and Prins extended DI to XY geometry. Both authors utilized a combination of analytic solutions and a traditional source iteration code as benchmarks. The suite of tests was extended beyond what was required for verification in order to evaluate performance and develop confidence in DI.

5.1.1 Slab Geometry. The performance of the time independent, slab geometry DI method was evaluated by Wager [32]. Wager examined two test cases, the first one was a single region problem and the second one was a periodic interface problem consisting of two regions repeated ten times. Building from his work, I implemented a series of tests designed to detect common programming errors and implementation failures. The set of test problems is presented in the following list–the details of which follow after the list (dimensions are in units of mean free path (MFP)):

1. Pure Absorber-Single region ($\sigma = 1, c = 0, q^{\mathrm{e}} = 4.7$, 32 mfp) and vacuum boundaries;
2. Pure Scatterer-Single region ($\sigma = 1, c = 1, q^{\mathrm{e}} = 4.7$, 32 mfp) and vacuum boundaries;
3. Infinite Medium-Single region ($\sigma = 1, c = 0.5, q^{\mathrm{e}} = 4.7$, 32 mfp), symmetry boundaries;
4. Constant Source-Single region ($\sigma = 0.1, c = 0.5, q^{\mathrm{e}} = 4.7$, 3.2 mfp) vacuum boundaries;
5. Left Isotropic Surface Source ($Q = 0.5$), 4 mfp-Single region ($\sigma = 1, c = 0.9, q^{\mathrm{e}} = 0$, 4 mfp), left symmetry boundary and right vacuum boundary;
6. Left Isotropic Surface Source ($Q = 0.5$), Infinite Medium-Single region ($\sigma = 1, c = 0.9, q^{\mathrm{e}} = 0$, Infinite), left symmetry boundary and right albedo matrix;
7. Left Lambertian Illumination ($Q = 4.7$), 4 mfp-Single Region ($\sigma = 1, c = 0.9, q^{\mathrm{e}} = 0$, 4 mfp), vacuum boundaries;

8. 1D Periodic Interface, Cross-sections: Region A ($\sigma = 10, c = 1.0, q^{\mathrm{e}} = 0$, 10 mfp) Region B ($\sigma = 0.1, c = 1.0, q^{\mathrm{e}} = 0$, 0.1 mfp), Left Lambertian illumination ($Q = 4.7$), vacuum boundaries;
9. 1D Periodic Interface, Scattering ratio: Region A ($\sigma = 1, c = 1, q^{\mathrm{e}} = 0$, 1 mfp) Region B ($\sigma = 1, c = 0, q^{\mathrm{e}} = 0$, 1 mfp), Left Lambertian illumination ($Q = 4.7$), vacuum boundaries;
10. 1D, Two Region, Scattering ratio test, Region A and B ($\sigma = 1.0, S = 0$, 2 mfp), vacuum boundaries, left incident Lambertian illumination.

These benchmark problems form the basis of the different test scenarios as shown in Table 5.1. In all cases, unless otherwise noted, the convergence criterion was that the symmetric relative difference in ζ between iterations was less than 1×10^{-7} everywhere. The symmetric relative difference is defined as

$$\mathrm{SRD}(x, y) = 2\frac{|x - y|}{|x| + |y|}. \tag{5.1}$$

The key verification metric that I evaluated was the maximum symmetric relative difference in the scalar flux computed by DI and the benchmark. If the maximum symmetric relative difference between the two solutions was less than 10^{-6}, then the two solutions were in agreement. I also examined the rate of convergence as the spatial mesh was refined for one of the test problems.

Because much of this work was performed by Wager, the focus of my effort was to verify that I had correctly implemented the DI algorithm rather than a detailed assessment of the algorithm. I did expand upon Wager's assessment of DI's performance by comparing the performance of DI with a convergence-accelerated source iteration code.

5.1.1.1 Tests 1 and 2. The single region pure absorber test (1) is the simplest possible test and provides a useful starting point for developing a transport code. The key advantage of this problem is the fact that it has an analytic solution,

Table 5.1: Summary of time independent, slab geometry tests.

Test Problem	Analytic	Exact	PARTISN
Pure Absorber	All		DD/LD
Pure Scatterer			DD/LD
Infinite Medium	All		DD/LD
Constant Source, Single-Region		SC/LC/LD	DD/LD
Isotropic Source, 4 mfp		SC	
Isotropic Source, Infinite Medium		SC	
Lambertian Source, 4 mfp, Vacuum			DD/LD
Periodic Interface, Cross-sections			DD/LD
Periodic Interface, Scattering ratio			DD/LD

which greatly facilitates the debugging of a transport code. The complement to the pure absorber test is the pure scatterer test (2). The primary utility of the pure scatterer test is that it tests a different part of the transport code, which is useful for debugging purposes. Using a spatial mesh of 256 uniformly spaced cells and the S_8 and DP_4 angular quadratures the solution produced by DI in both tests agreed with the benchmark solutions.

I wanted to determine if there was a difference in accuracy between the single-range and double-range angular quadratures. Carlson [8] asserts that the double-range angular quadrature yields more accurate results for thin cells than the single-range angular quadrature. Conversely, the single-range angular quadrature yields more accurate results for thick cells because there is a higher probability of neutrons traveling out of a cell in directions near $\mu = \pm 1$ and the single-range angular quadrature has ordinates closer to $\mu = \pm 1$ than has the double-range angular quadrature.

To determine if there was such a difference between the two angular quadratures, I used a variant of the pure absorber test. Instead of a uniform source, I used a Lambertian illumination on the left boundary. The scalar flux on a cell face for a

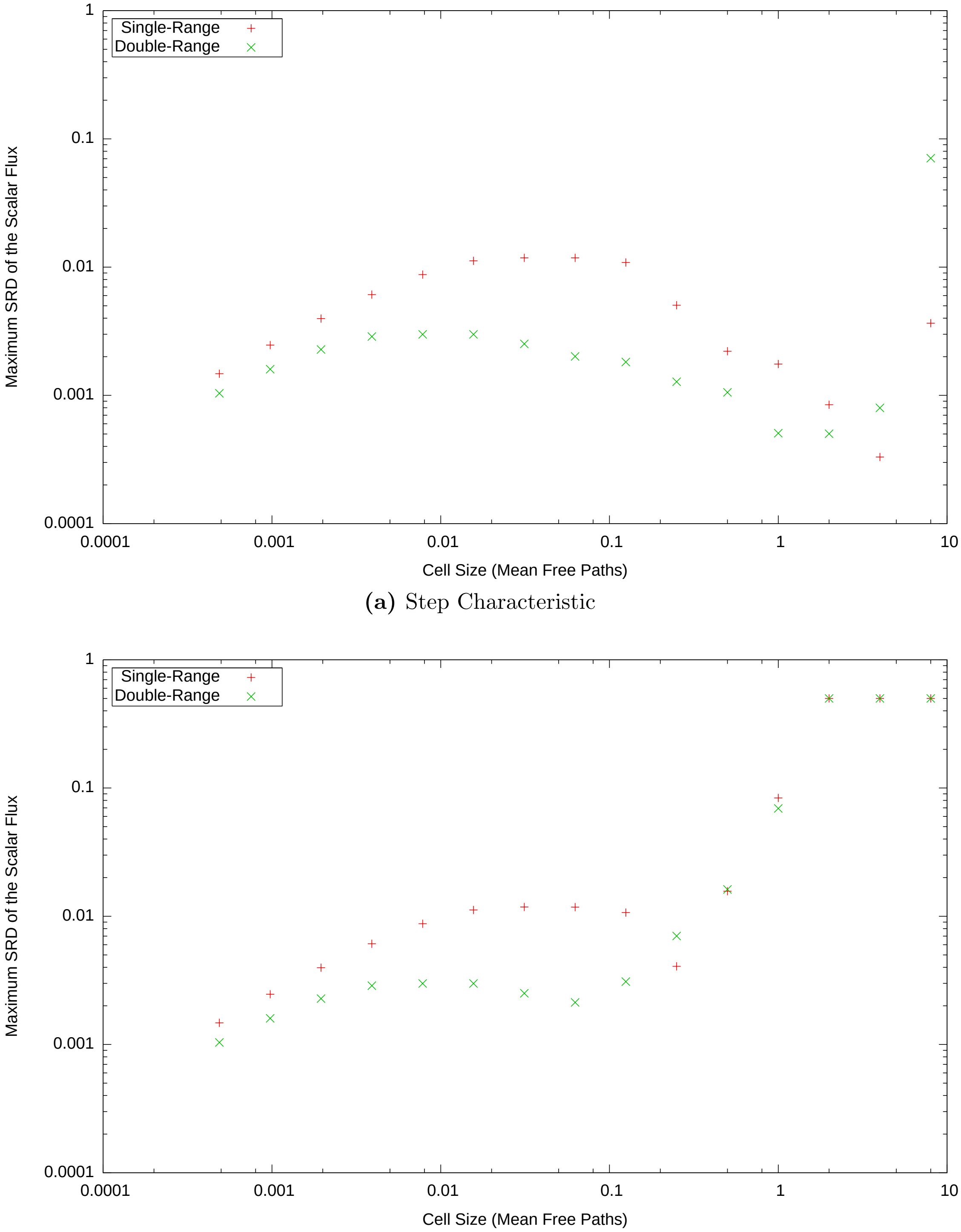
Lambertian illumination is simply

$$\phi(x) = \frac{Q}{2}\left(e^{-x\sigma} - x\sigma\Gamma(0, x\sigma)\right), \tag{5.2}$$

where $\Gamma(0, x\sigma)$ is the upper incomplete Gamma function. Using the S_8 and DP_4 angular quadratures, I evaluated the symmetric relative difference between the resulting cell face scalar fluxes and the exact solution. When using step characteristic, the single-range angular quadrature did perform better when the cell size was greater than 3 mean free paths (Figure 5.1). The linear discontinuous spatial quadrature, which is not a positive spatial quadrature, did not exhibit as large of a performance difference between the two angular quadratures. In fact, both angular quadratures were ineffective when the cell size was large enough to cause negative fluxes (> 2 mfp). For both spatial quadratures, the double-range angular quadrature did have superior accuracy for the thinner spatial cells, as asserted by Carlson.

5.1.1.2 Test 3. The infinite medium test (3) serves as an intermediate test between the pure absorber and pure scatterer tests. The use of left and right symmetry boundaries, as well as having $c \neq 1$, yields an analytic solution for the scalar flux: $\phi = S/\sigma_a$. Using a spatial mesh of 256 uniformly spaced cells and the S_8 and DP_4 angular quadratures the solutions produced by DI agreed with the benchmark solutions.

5.1.1.3 Test 4. The constant source, single region problem (4) served as the basis for evaluating the convergence rate. Moderate scattering ($c = 0.5$) was chosen to permit rapid convergence of unaccelerated source iteration. The spatial mesh was refined from 4 cells to 65536 cells and the test was performed using the various spatial quadratures (SC, DD, LD, and LC). Both single-range and double-range Gauss-Legendre angular quadratures were used with 8 ordinates (S_8 and DP_4) and the convergence tolerance was set to 10^{-7}. Using a spatial mesh of 256 uniformly



(a) Step Characteristic

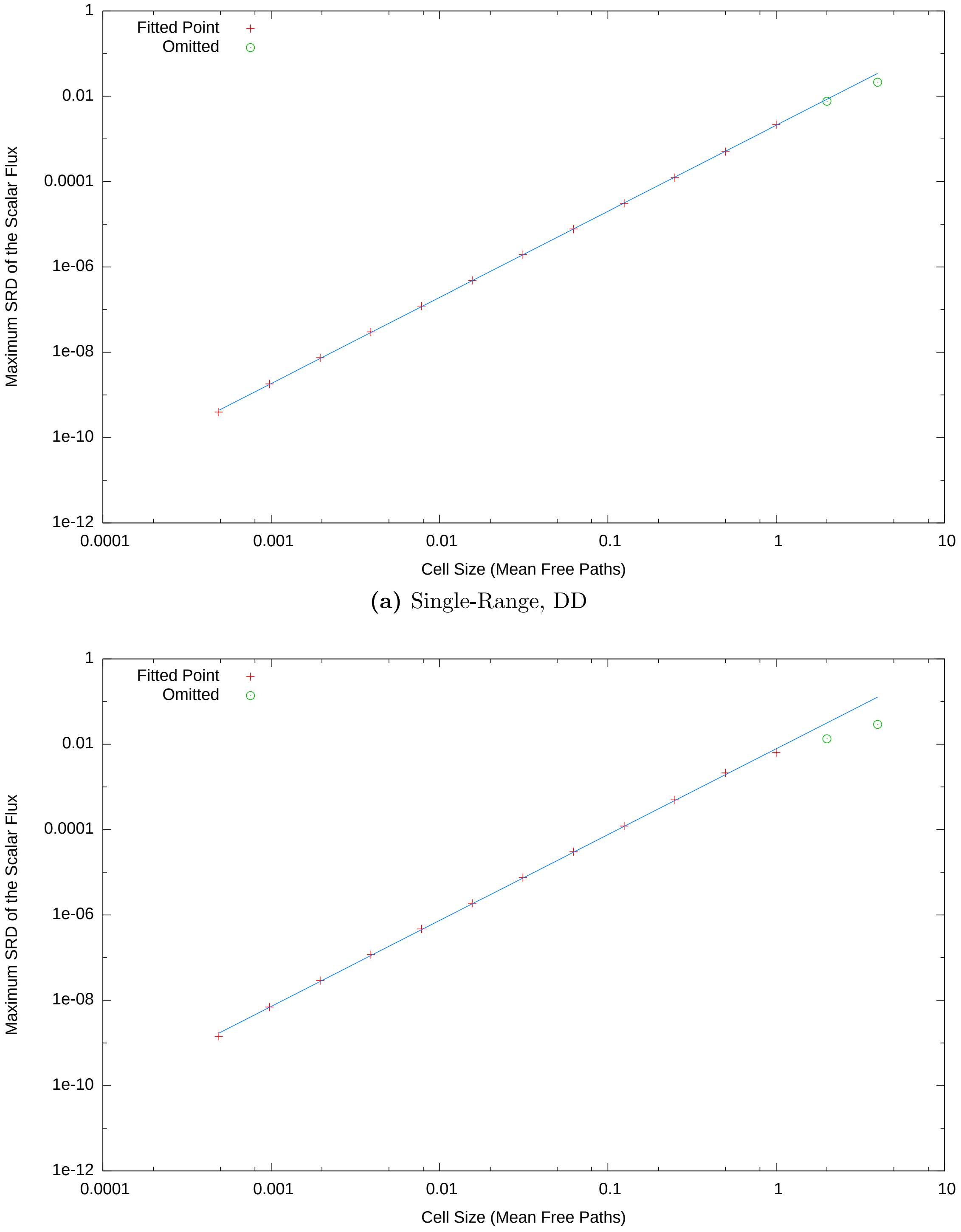
(b) Linear Discontinuous

Figure 5.1: Comparison of the angular quadratures for the step characteristic and linear discontinuous spatial quadratures.

spaced cells and the S_8 and DP_4 angular quadratures the solutions produced by DI agreed with the benchmark solutions.

The convergence rate for each spatial and angular quadrature combination was determined by computing the maximum symmetric relative difference between the cell face scalar flux for each cell width and the corresponding baseline solution. The baseline solution for each combination was computed using a cell width of 4.88×10^{-6}. The observed rate of convergence for the different spatial quadratures for both the S_8 and DP_4 angular quadratures agrees well with the analytical expectations from Larsen [19] (Table 5.2). The LD and LC spatial quadratures are of particular interest because of their high rate of convergence. Larsen demonstrated that the global discretization error for LD and LC was $O(\Delta x^2)$ and $O(\Delta x^3)$, respectively. He further demonstrated that LD and LC will exhibit a "superconvergence" in the cell-averaged angular fluxes. The rate of convergence in the cell-averaged angular fluxes will be one higher than the global discretization error. Furthermore, Larsen demonstrated that the superconvergence of LD and LC also applies to cell-edge angular fluxes. Figures 5.2 to 5.5 show, for each of the four spatial quadratures, the maximum value of the symmetric relative difference along with the convergence rate produced using a linear fit. For LD and LC, some values were omitted when the cell size was small because the solution had converged to the benchmark. For optically thick cells, some values were omitted because there was additional sources of error, e.g. DD had negative fluxes for very optically thick cells.

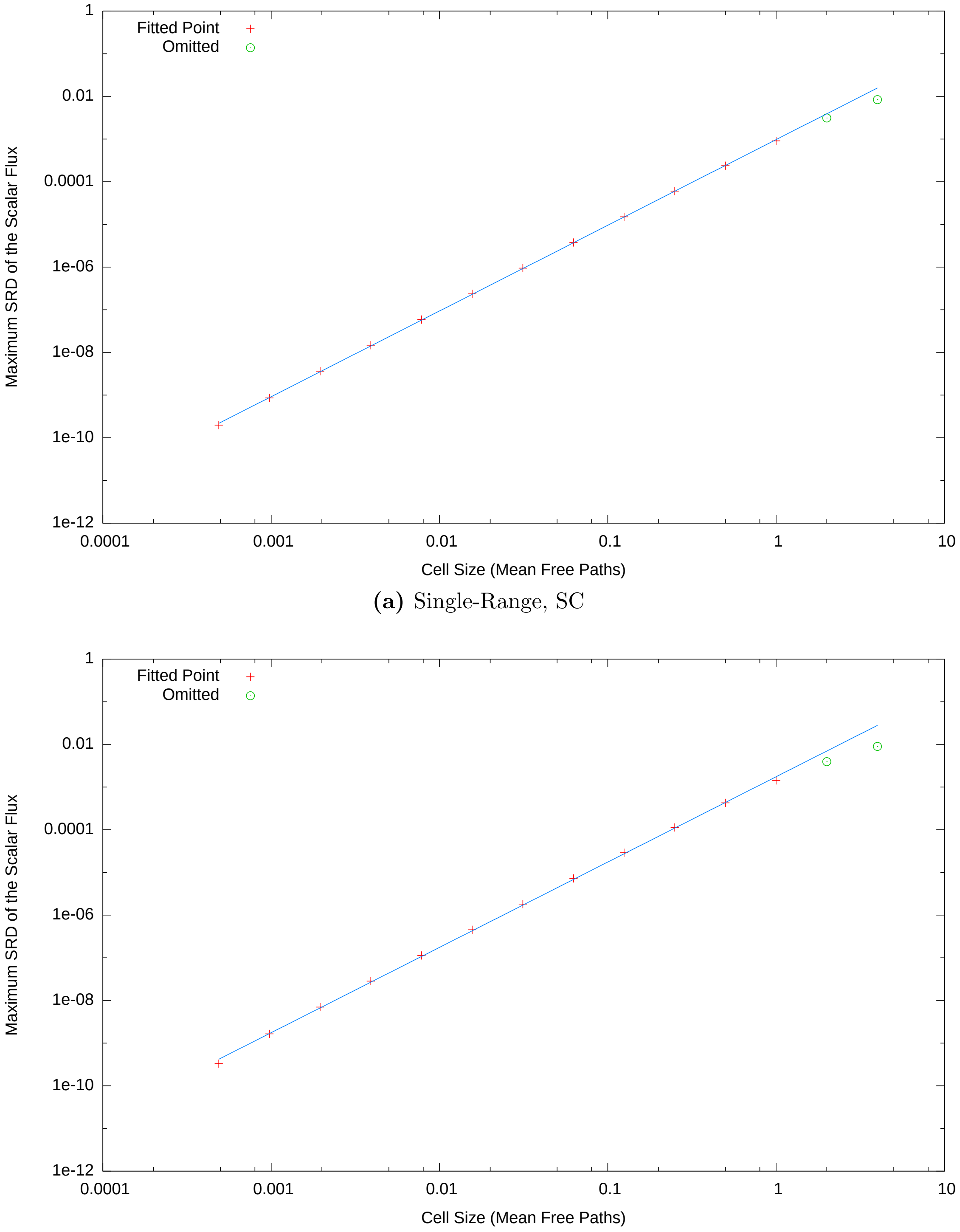
When this test was performed with a double-range Gauss-Legendre angular quadrature and the total number of ordinates was still 8 (DP_4), the number of iterations required for convergence was unchanged. Note that, for optically thick cells, the omitted values for the single-range angular quadrature results were closer to the convergence rate line than the double-range angular quadrature results. The converse is true for optically thin cells. This effect is the manifestation of the performance difference of the angular quadratures in optically thick versus optically thin cells mentioned previously.



(a) Single-Range, DD

(b) Double-Range, DD

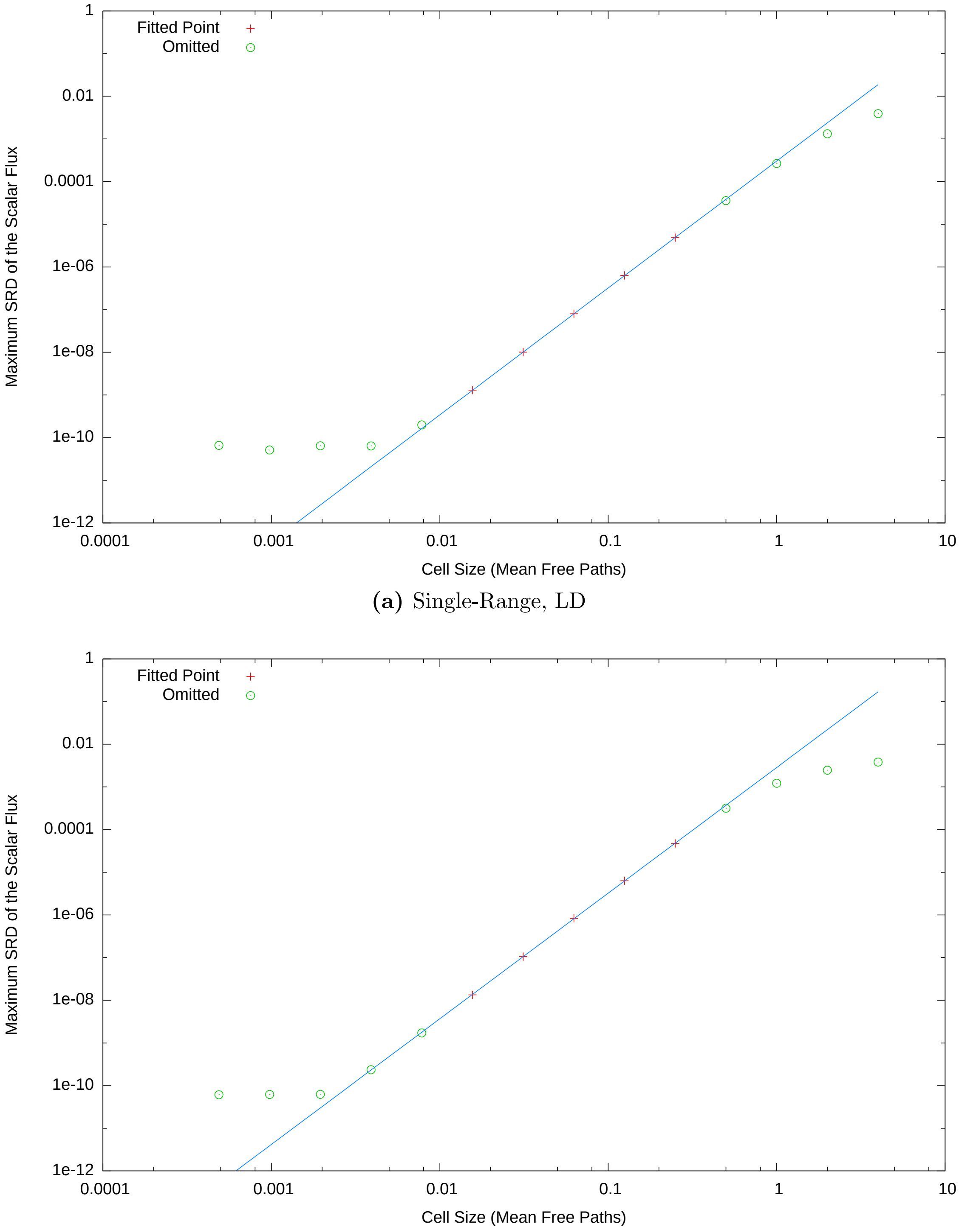
Figure 5.2: Spatial mesh refinement convergence rate in slab geometry using Distribution Iteration and diamond difference.



(a) Single-Range, SC

(b) Double-Range, SC

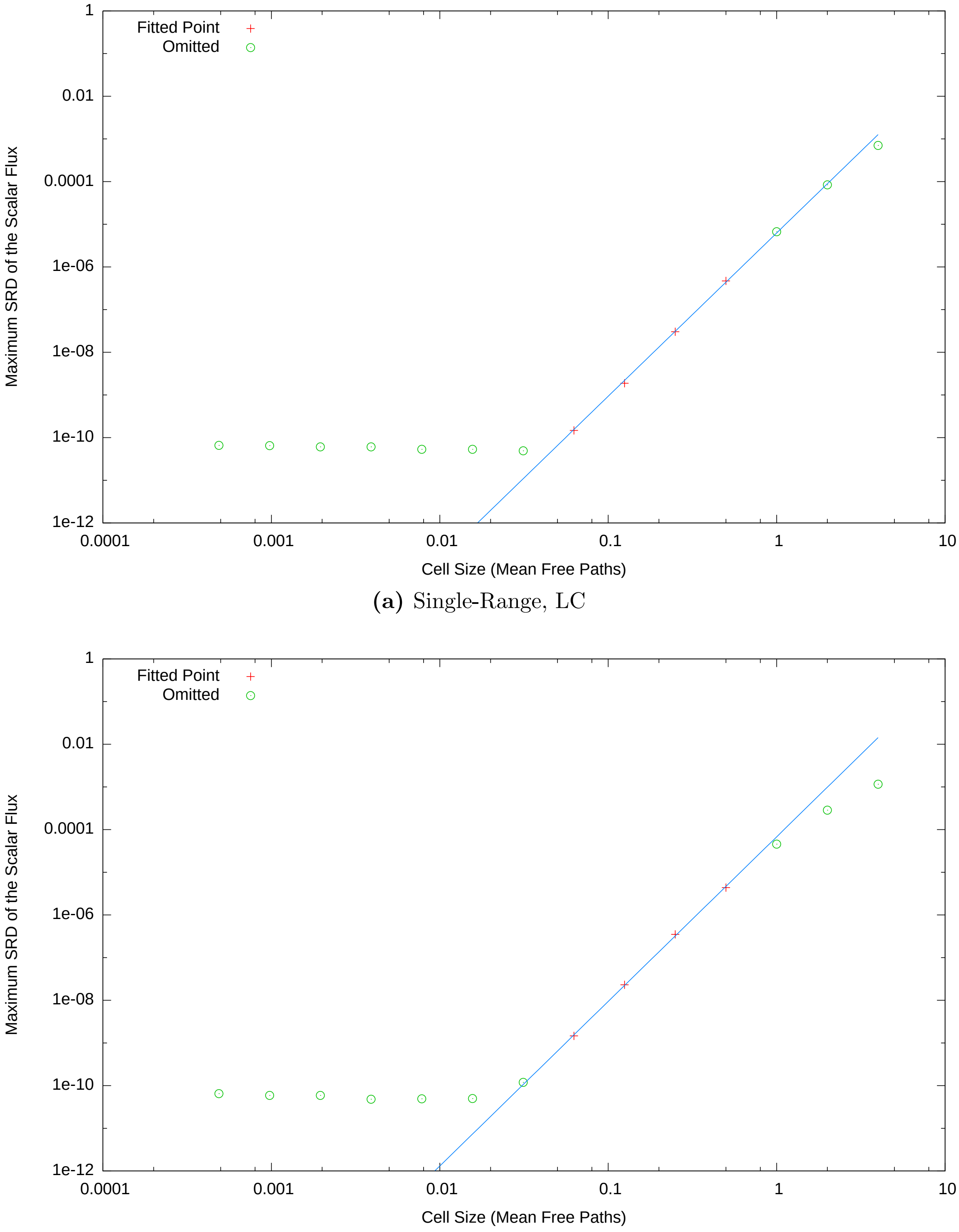
Figure 5.3: Spatial mesh refinement convergence rate in slab geometry using Distribution Iteration and step characteristic.



(a) Single-Range, LD

(b) Double-Range, LD

Figure 5.4: Spatial mesh refinement convergence rate in slab geometry using Distribution Iteration and linear discontinuous.



(a) Single-Range, LC

(b) Double-Range, LC

Figure 5.5: Spatial mesh refinement convergence rate in slab geometry using Distribution Iteration and linear characteristic.

Table 5.2: Convergence rates for the constant source, single region problem using the Distribution Iteration algorithm.

	Convergence Rate		
Spatial Quadrature	Single-Range	Double-Range	Analytic
DD	2.0	2.0	2
SC	2.0	2.0	2
LD	3.0	2.9	3
LC	3.8	3.9	4

5.1.1.4 Tests 5 and 6. The two problems with the left isotropic surface source (5 and 6) were developed to exploit Ganapol's TIEL benchmark. Both the 4 mfp and the infinite medium tests were also used to test left/right symmetry. Using a spatial mesh of 2048 uniformly spaced cells and the S_{32} and DP_{16} angular quadratures, the infinite medium test was compared with the TIEL benchmark and the scalar flux was within the convergence tolerance except when too close to the source. The agreement of the scalar flux in proximity to the isotropic surface source was dependent upon the angular quadrature order; the higher the order the closer one could approach the source.

5.1.1.5 Test 7. Ideally, the isotropic test would also be used with the source iteration code, however this was not possible because PARTISN does not permit a symmetry boundary and an incident current to share a surface. Instead I developed a separate test using a Lambertian illumination of the left boundary and left and right vacuum boundaries for comparing DI with the source iteration code. The change from isotropic surface source to the Lambertian illumination was due to the need to test the implementation of that source type within the DI test code. Using a spatial mesh of 512 uniformly spaced cells and the S_8 and DP_4 angular quadratures both DI and the source iteration solutions agreed to within the agreement tolerance.

5.1.1.6 Tests 8 and 9. The periodic interface problems (8 and 9) are used to assess performance in regimes where there is a repeated change in material properties. These tests were inspired by the periodic horizontal interface problem presented by Azmy [5]. Even though Azmy's test was designed for XY geometry, it does pose an interesting test case in slab geometry because it provides a mean to study the how alternating material interfaces and deep penetration affect the performance of DI.

Test 8 creates a strong discontinuity in total cross-section between pure-scattering regions. In the high cross-section regions there is a strong scattering source causing an isotropic angular flux being transported to the adjacent cells. The low cross-section regions there is a weak scattering source, which results in a near streaming transport of the inward fluxes to the adjacent cells.

These tests were conducted by growing the problem by adding a *region pair*, which is just the combination of the two regions A and B. Using a spatial mesh of 64 uniformly spaced cells for each region and the S_8 and DP_4 angular quadratures, the maximum value of the symmetric relative difference was within the convergence tolerance when using diffusion synthetic acceleration (DSA); however, for unaccelerated source iteration and transport synthetic acceleration (TSA), the difference was larger than the convergence tolerance (Figure 5.6). Note that PARTISN turned off DSA after four region pairs because it was no longer effective. The reason for the lack of agreement between DI and the solutions produced by SI is due to false convergence on the part of SI. When DSA was enabled, the solutions produced by DI and SI were in agreement. When DSA was disabled at five region pairs, the two solutions were no longer in agreement.

Test 9 causes an alternating change in the angular distribution of the flux. The $c = 1$ regions produce an outgoing isotropic angular flux distribution, while the $c = 0$ regions produce an angular flux distribution that is biased towards the X-axis. The combination of the scattering regions, which effectively divides the flux of

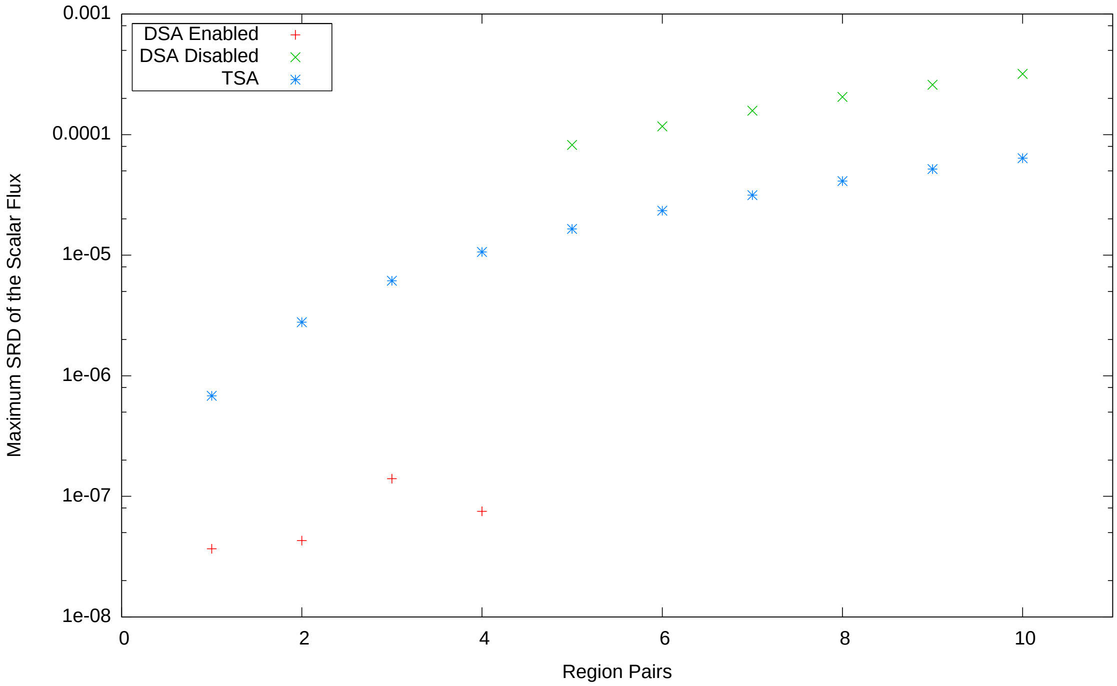


Figure 5.6: Symmetric Relative Difference for the Periodic Interface, Cross-Section Problem Using Diamond-Difference and Single-Range

neutrons into leftward and rightward flows, and the absorbing regions results in a diminishing neutron flux. This provides an opportunity to evaluate the performance in deep penetration problems. Using a spatial mesh of 16 uniformly spaced cells for each region and the S_8 and DP_4 angular quadratures, the agreement between DI and PARTISN was within the convergence tolerance to about 28 (unaccelerated) to 33 (DSA) region pairs. Again, the lack of agreement between the DI solutions and SI solutions is due to false convergence by SI.

5.1.1.7 Test 10. The two region scattering ratio test provides insight into the performance of a transport code for a variety of material type interfaces and was performed with both DI and PARTISN. Furthermore, the source iteration code was run with no source acceleration, with DSA, and with TSA. The preferred angular quadrature for an incident current problem is double-range Gauss-Legendre

(DP_n) because of its superior performance in solving for cell face angular fluxes[2] and was used in all cases except with TSA. PARTISN does not support the use of DP_n with TSA; therefore only the traditional S_n quadrature was used with TSA. Using a spatial mesh of 16 uniformly spaced cells for each region and the S_8 and DP_4 angular quadratures, the scalar fluxes calculated by DI in all cases were within the convergence tolerance of the solutions generated by PARTISN.

5.1.2 XY Geometry. Prins [28] developed and examined the performance of DI in XY geometry. I implemented a series of test cases similar to the ones used in slab geometry. The set of test problems is presented in the following list:

1. Pure Absorber-Single region ($\sigma = 1, c = 0, q^{\mathrm{e}} = 4.7$, 32 mfp by 32 mfp) and vacuum boundaries;
2. Pure Scatterer-Single region ($\sigma = 1, c = 1, q^{\mathrm{e}} = 4.7$, 32 mfp by 32 mfp) and vacuum boundaries;
3. Infinite Medium-Single region ($\sigma = 1, c = 0.5, q^{\mathrm{e}} = 4.7$, 32 mfp by 32 mfp), symmetry boundaries;
4. Constant Source-Single region ($\sigma = 0.1, c = 0.5, q^{\mathrm{e}} = 4.7$, 3.2 mfp by 32 mfp by 32 mfp), vacuum boundaries;
5. Left Isotropic Surface Source ($Q = 0.5$), 4 mfp-Single region ($\sigma = 1, c = 0.9, q^{\mathrm{e}} = 0$, 4 mfp by 4 mfp), left/top/bottom symmetry boundaries and right vacuum boundary;
6. Left Isotropic Surface Source ($Q = 0.5$), 32 mfp-Single region ($\sigma = 1, c = 0.9, q^{\mathrm{e}} = 0$, 32 mfp by 4 mfp), left/top/bottom symmetry boundaries and right vacuum boundary;
7. Left Lambertian Illumination ($Q = 4.7$), 4 mfp-Single Region ($\sigma = 1, c = 0.9, q^{\mathrm{e}} = 0$, 4 mfp by 4 mfp), vacuum boundaries;

[2] The S_n angular quadrature can be off by 1% or more for thin cells

The only spatial quadrature in XY geometry in common between PARTISN and my DI code was diamond difference. Also, all comparisons were done using the S_4 angular quadrature. The grid size in all test cases was set to 20 by 20; the only exception being the left isotropic surface source test case which used a 40 by 40 for the 4 mfp case and a 320 by 40 for the 32 mfp case.

The use of the diamond difference spatial quadrature is problematic in XY geometry as it is possible to generate negative fluxes. While there are flux fix up methodologies, there are drawbacks to using them. I deliberately decided to use diamond difference in the verification effort because of the negative flux problem. I wanted to evaluate the performance of DI when negative fluxes are generated as this should make it difficult for DI to converge. In addition to diamond difference, I implemented the step characteristic and linear characteristic spatial quadratures.

I did not perform any spatial mesh or angular quadrature refinement testing in the XY geometry. The structure of my DI code in the XY and XYZ geometries was designed to facilitate debugging and to provide a detailed look at how the solution is progressing. As a result, it is neither memory nor computationally efficient. Also, the implementation of the linear discontinuous spatial quadrature is necessary to provide more reliable results that can be compared with PARTISN.

The tests in which both DI and PARTISN (unaccelerated and DSA) agree to within the convergence tolerance are:

- Pure Absorber (Test 1)
- Infinite Medium (Test 3)
- Constant Source, Single-Region (Test 4)
- Lambertian Source, 4mfp, Vacuum (Test 7)

In the pure scatterer test (Test 2), only DSA agreed to within the convergence tolerance. Unaccelerated source iteration required 586 iterations and did not agree with either DI or with DSA. This difference is due to false convergence by unaccelerated SI.

The isotropic surface source tests were used to compare with the TIEL benchmark. The matrix albedo was not implemented in XY nor XYZ geometry, thus a 32 mfp problem was used to emulate an infinite medium. When using diamond difference, DI was unable to generate a reasonable solution due to the negative flux effect. The step characteristic spatial quadrature was much closer–the maximum symmetric relative difference was 3.0×10^{-2} for optical depths in excess of 1 mfp. Given the performance constraints of my DI code due to the way it was implemented, I did not refine the spatial and angular mesh to refine the solution.

The two region, scattering ratio sweep was implemented in XY geometry as two slabs with dimensions 2 mfp by 4 mfp (Figure 5.7), with vacuum boundaries on all four sides. A Lambertian illumination was used on the left boundary. Of the hundred combinations of scattering ratios, there were eight cases where DI and DSA did not agree; in three of the eight cases, DI had failed to converge within 500 iterations. This will be discussed further in Section 6.1.2.

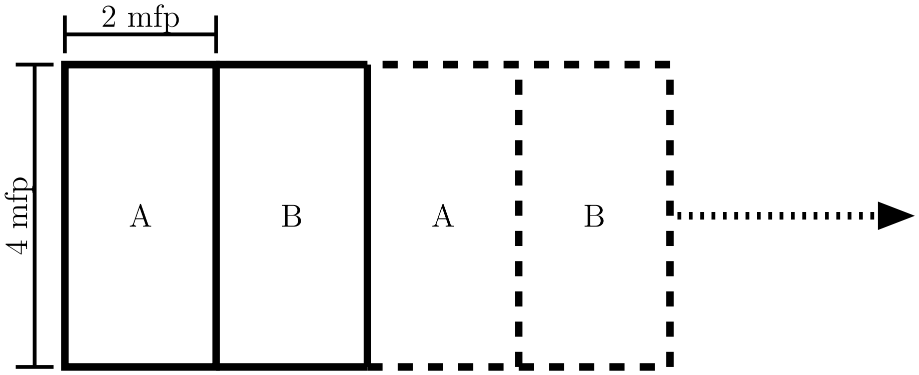


Figure 5.7: Layout for the two region test.

5.1.3 XYZ Geometry. Because XYZ geometry was an extension of the XY implementation, a limited set of test cases were required. The focus of the testing was to ensure that Z axis currents and the spatial quadratures were implemented correctly. The set of test problems performed in XYZ geometry is presented in the following list:

1. Pure Absorber-Single region ($\sigma = 1, c = 0, q^{\mathrm{e}} = 4.7$, 32 mfp by 32 mfp by 32 mfp) and vacuum boundaries;
2. Pure Scatterer-Single region ($\sigma = 1, c = 1, q^{\mathrm{e}} = 4.7$, 32 mfp by 32 mfp by 32 mfp) and vacuum boundaries;
3. 1D, Two Region, Scattering ratio test ($\sigma = 1.0, q^{\mathrm{e}} = 0$, 2 mfp by 4 mfp by 4 mfp), vacuum boundaries, left incident Lambertian illumination ($Q = 4.7$).

The spatial mesh in all cases was a 20 by 20 by 20 uniform cell spacing using the S_4 angular quadrature.

While using DD with DI in XY geometry caused some problems in converging, in XYZ geometry DI exhibited exceptionally poor performance when using diamond difference. For problems with strong emission sources, such as the pure absorber and pure scatterer tests, DI and PARTISN converged to the same solutions. Using a positive method, such as step, DI is able to converge for all the test cases. My negative flux fix-up scheme was able to fix many of the cases were DI did not converge, however, the scheme failed when the scattering ratio was larger than 0.8.

5.2 Multigroup Verification Tests

Three multigroup test cases were implemented to verify that the conventional multigroup method is compatible with the DI method. The first test was an absorptive downscatter-only problem utilizing three energy groups. The group cross sections are shown in Table 5.3, where σ_a is the absorption cross-section, σ_{t} is the total cross-section, and $\sigma_{\mathrm{s}_{g\to 2}}$ is the scattering cross-section from group g to group 2. The downscatter only test is the simplest multigroup problem and it is a useful test for debugging purposes. The next test was an absorptive upscatter test. The group cross sections are similar to those shown in Table 5.3, the only change being that the last group has an upscatter contribution to the first group. The third test was a downscatter only test where the second energy group was a strong scatterer (Table 5.4). This test was designed to have a different inward angular partial current distribution between adjacent energy groups. The alternating pattern would test how well DI is able to adapt to the change in the inward angular current distribution.

In slab geometry, all three test cases consisted of a 32 mfp uniform region with vacuum boundaries and a 256 uniform spatial cell mesh. Both the S_8 and DP_4 angular quadratures were used. In XY geometry a 32 mfp by 32 mfp uniform region and 20 by 20 uniform spatial cell mesh was used. The S_4 angular quadrature was used for XY geometry. For the upscatter test, the convergence test was taking the maximum symmetric relative difference between the current and previous iteration for each energy group. The convergence tolerance was set to 10^{-7}.

DI and PARTISN (unaccelerated and TSA) both agreed using all combinations of angular quadratures and supported spatial quadratures (LD and DD in slab geometry and DD in XY geometry). PARTISN did not agree to within the convergence tolerance on the "Three Group Absorber Upscatter Test" when using DSA in both slab and XY geometries. When using DSA, the maximum value of the symmetric relative difference for the different combinations of spatial and angular quadratures was 7×10^{-6} to 9×10^{-6}, just beyond the agreement requirement of 10^{-6}.

5.3 Time Dependent Verification Tests

Three time-dependent external sources, q^{e}, were used for testing. The first source was a uniform isotropic source that is initially off and instantaneously turns

Table 5.3: Group cross sections use for the downscatter test.

Energy Group	σ_a	σ_{t}	$\sigma_{\mathrm{s}_{g\to 1}}$	$\sigma_{\mathrm{s}_{g\to 2}}$	$\sigma_{\mathrm{s}_{g\to 3}}$
1	0.9	1.0	0.0	0.1	0.0
2	0.9	1.0	0.0	0.0	0.1
3	1.0	1.0	0.0	0.0	0.0

Table 5.4: Group cross sections use for the alternating test.

Energy Group	σ_a	σ_t	$\sigma_{\mathrm{s}_{g\to 1}}$	$\sigma_{\mathrm{s}_{g\to 2}}$	$\sigma_{\mathrm{s}_{g\to 3}}$
1	0.9	1.0	0.0	0.1	0.0
2	0.0	1.0	0.0	0.9	0.1
3	1.0	1.0	0.0	0.0	0.0

on and then stays constant in time (Step Source). The "Step Source" test is the most rudimentary time dependent test and is useful in debugging. This test was also used to verify that the solution was approaching the steady-state solution. The second and third sources were a "Ramp Up and Hold" and a "On and Ramp Down." The combination of these three tests would test the performance of the DI method for the common time-dependent source types. In all test case a single energy group was used with a velocity of 10^{-3}cm/s.

Comparing PARTISN and DI on a time step basis was not feasible because of the adaptive time step algorithm that PARTISN uses. I decided that implementing PARTISN's adaptive time step algorithm was not useful nor was modifying PARTISN to create a file containing the time steps. Instead, I had DI use a smaller time step, 10^{-4} seconds, and interpolated between time steps in order to compare with PARTISN. The basis for my decision was that PARTISN's utility as an accurate benchmark is limited by its inability to output the more accurate extrapolated fluxes (see Appendix A). Furthermore, the focus of my research was to demonstrate that DI did not have problems converging when used with very optically thick spatial cells. The time-independent testing demonstrated that DI produces reliable results when the spatial cell is the appropriate size for the spatial quadrature.

The synthetic cross section for the test cases performed by DI was 10^7 cm^{-1}. For PARTISN, the synthetic cross section varied between 10^7 cm^{-1} and 10^4 cm^{-1}. The constant source test was also used to test a range of time steps, ranging from 10 seconds to 10^{-5} seconds.

Table 5.5 shows the maximum and mean values for the symmetric relative difference between the DI and PARTISN solutions taken at each PARTISN time step. The interpolation was performed using Mathematica's interpolation function, which performs a piecewise polynomial fit of order 3. This test was performed when DI was set to output the midpoint scalar flux and the more accurate end of time step scalar flux. There was no significant difference when the time step was varied.

For XY geometry, a corner source problem was used. The problem was 6 cm by 6 cm with an uniform isotropic source region in the lower left corner with dimensions 1 cm by 1 cm. The source region material had a cross section of 0.1cm^{-1} and a scattering ratio of 0.5. The rest of the problem was a uniform material with a cross section of 1.0cm^{-1} and a scattering ratio of 0.5. A 20 by 20 spatial mesh was used and the angular quadrature was S$_4$.

In XY geometry, comparing the results generated by PARTISN and DI was problematic. When using DD as the spatial quadrature a large number of negative scalar fluxes will occur outside the source region. PARTISN forces a negative flux fix-up for time dependent problems, thus it is not possible to compare my DI code and PARTISN without any fix-up. Because I did not implement the same negative flux fix-up, there is a substantial difference, a symmetric relative difference of approximately 0.1, in the results generated by DI and PARTISN.

Table 5.5: Maximum and mean symmetric relative difference between DI and PARTISN for the time dependent tests in slab geometry.

Test	Maximum	Mean
Constant Source	1.0×10^{-4}	2.2×10^{-6}
On and Ramp Down	6.4×10^{-4}	6.3×10^{-4}
Ramp Up and Hold	6.0×10^{-4}	8.4×10^{-5}

VI. Performance Analysis

The verification testing that I performed demonstrated that DI converges to the benchmark solution in most of the test cases. An assessment of how the DI algorithm performed in reaching the solution is necessary in order to judge if DI is a suitable replacement for source iteration. The key performance metric in this assessment is the number of iterations required for convergence. Because of the different performance trade-offs made in my DI code, which is designed to support research, versus PARTISN, which is designed to be a production code, a run time or processor time comparison is not a reliable measure of performance.

In the case of PARTISN, the number of inner iterations is the measure of its performance. For DI, the number of calls to the partial current solver is the measure of its performance. In addition, an examination of when DI did not converge, or converged to a different solution, is part of the qualitative assessment of the overall performance of the DI method.

6.1 Time Independent Problems

All the test cases referenced in this chapter are described in Chapter V.

6.1.1 Slab Geometry. A summary of the number of iterations required for convergence is shown in Table 6.1. In all cases DI was able to converge to the benchmark solution in the same number or fewer iterations as PARTISN using DSA. Furthermore, with few exceptions, the agreement between the scalar fluxes produced by DI and PARTISN was within the convergence tolerance. The few exceptions where the two methods did not agree are presented later in the text. The isotropic source tests (5 and 6) was not performed with PARTISN because the source could not be accurately represented. The ranges given for tests 8 and 9 reflect the number of iterations from the smallest problem to the largest problem.

6.1.1.1 Tests 1 and 2. The single region pure absorber test (1) demonstrates that there is no performance advantage between DI and source iteration for

Table 6.1: Number of iterations required for convergence in the time independent,slab geometry tests.

		Source Iteration		
Benchmark	DI	None	DSA	TSA
1. Pure Absorber	2	2	2	2
2. Pure Scatterer	7	103	10	25
3. Infinite Medium	1	27	10	N/A
4. Constant Source, Single-Region	6	20	7	7
5. Isotropic Source, 4 mfp	11	N/A	N/A	N/A
6. Isotropic Source, Infinite Medium	11	N/A	N/A	N/A
7. Lambertian Source, 4 mfp, Vacuum	8	75	15	18
8. Periodic Interface, Cross-sections	9 - 12	540 - 27891	22 - 41684	123 - 6605
9. Periodic Interface, Scattering ratio	8 - 17	33 - 98	11 - 22	12 - 29

strong absorbing problems. As the scattering ratio increases (Test 4 and becomes a pure scattering problem (Test 2), the performance of unaccelerated source iteration drops. There is no performance advantage between DI and accelerated source iteration, either DSA or TSA, for problems that feature uniform material properties.

6.1.1.2 Test 3. This test problem is solved in one iteration by DI because the initial inward current distribution is isotropic, which is correct distribution for this problem. If the starting inward current distribution is set randomly, DI requires eight iterations to converge. This test suggests a useful performance enhancement for DI by using an approximation for the inward current distribution. This is particularly useful in a time-dependent code because in a slowly-varying problem the current distribution will not change rapidly between time steps. DI performed slightly better than DSA with this problem.

6.1.1.3 Test 4. Generally, all four spatial quadratures converged within 6 iterations, the only exceptions being at the coarse grid spacings where LC, SC, and DD required 5 iterations (SC required only 4 iterations at the coarsest mesh of 8 cells). As previously noted, there was no performance advantage between DI and accelerated source iteration. DI performed much better than unaccelerated source iteration.

6.1.1.4 Test 7. The Lambertian illumination problem exhibited a clear performance advantage between DI and accelerated source iteration. DI was able to converge in half as many iterations as DSA. Figures 6.1 through 6.4 show the number of iterations required for convergence as the scattering varied from 0 to 1 when using DSA, TSA, unaccelerated PARTISN and DI. Two test scenarios are used: One is a uniform source and the other is a Lambertian illumination. For DSA, Figure 6.1, there is a difference in the number of iterations between the two tests when $c > 0.2$. For TSA, Figure 6.2, the difference in the number of iterations does not occur until $c > 0.7$. Unaccelerated source iteration (Figure 6.3) is similar to TSA in behavior. While DI (Figure 6.4) does exhibit the difference in the number of iterations, there is no growth as $c \to 1$.

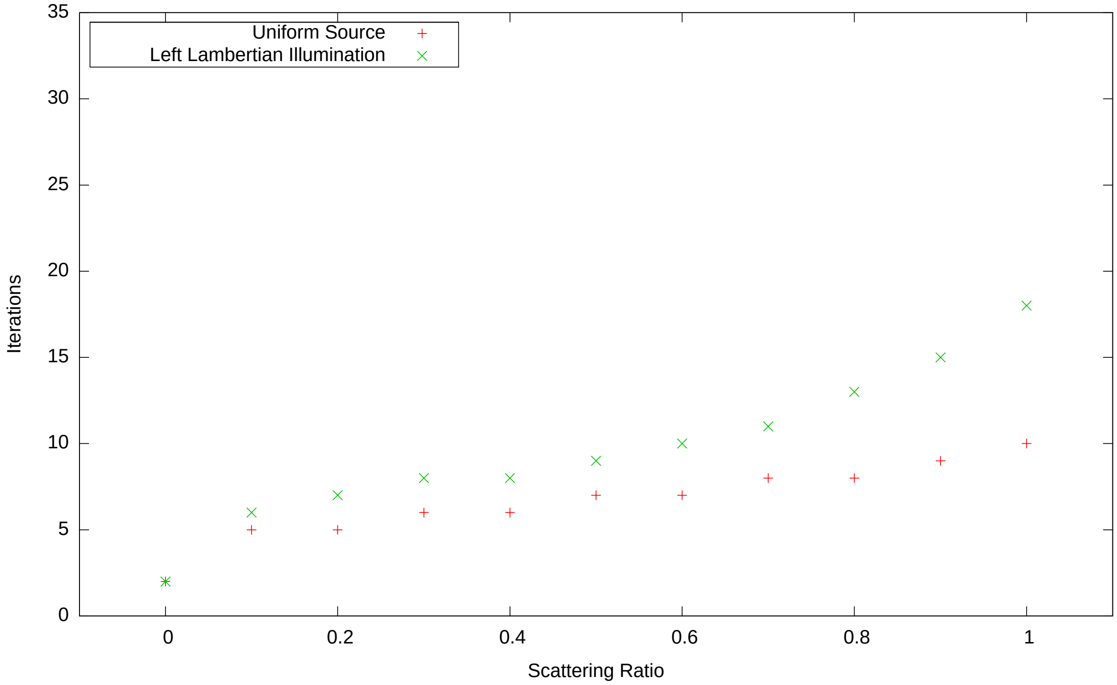


Figure 6.1: Number of iterations needed for Convergence using PARTISN with DSA, LD and S_8.

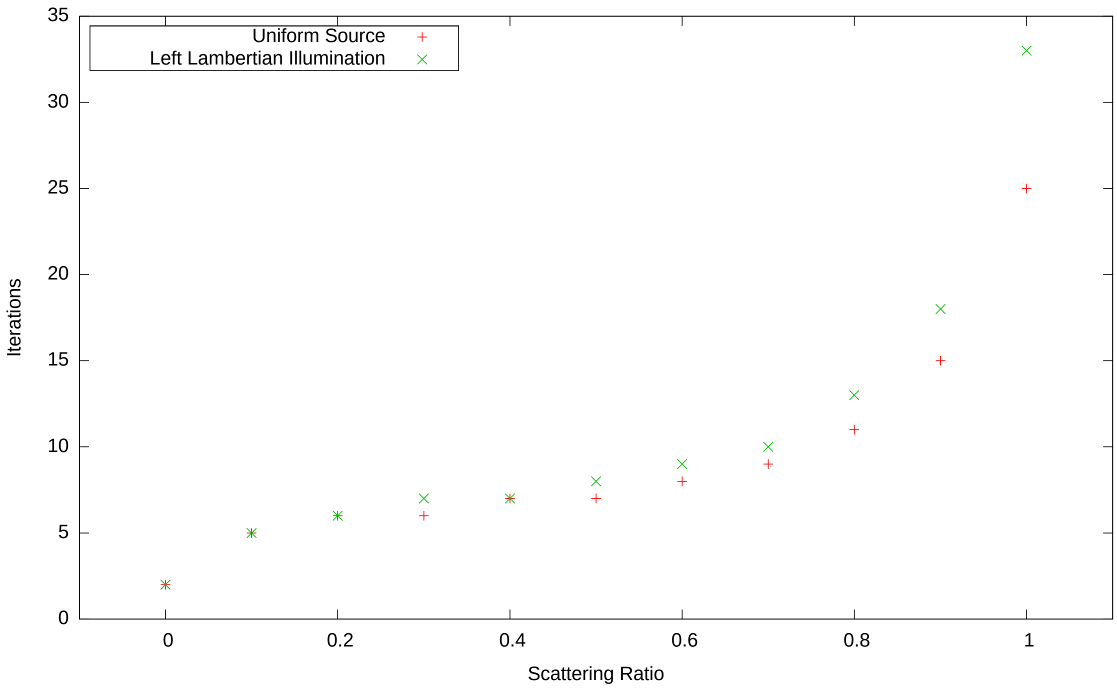


Figure 6.2: Number of iterations needed for convergence using PARTISN with TSA, LD and S_8.

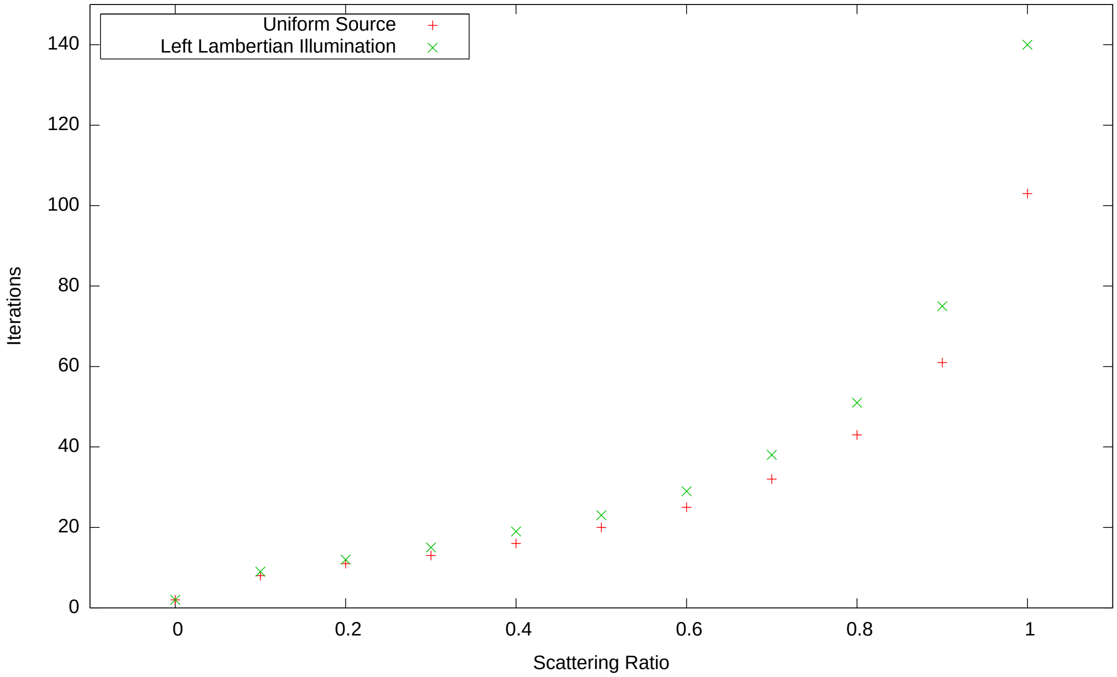


Figure 6.3: Number of Iterations Needed for Convergence Using Unaccelerated PARTISN, LD and S_8.

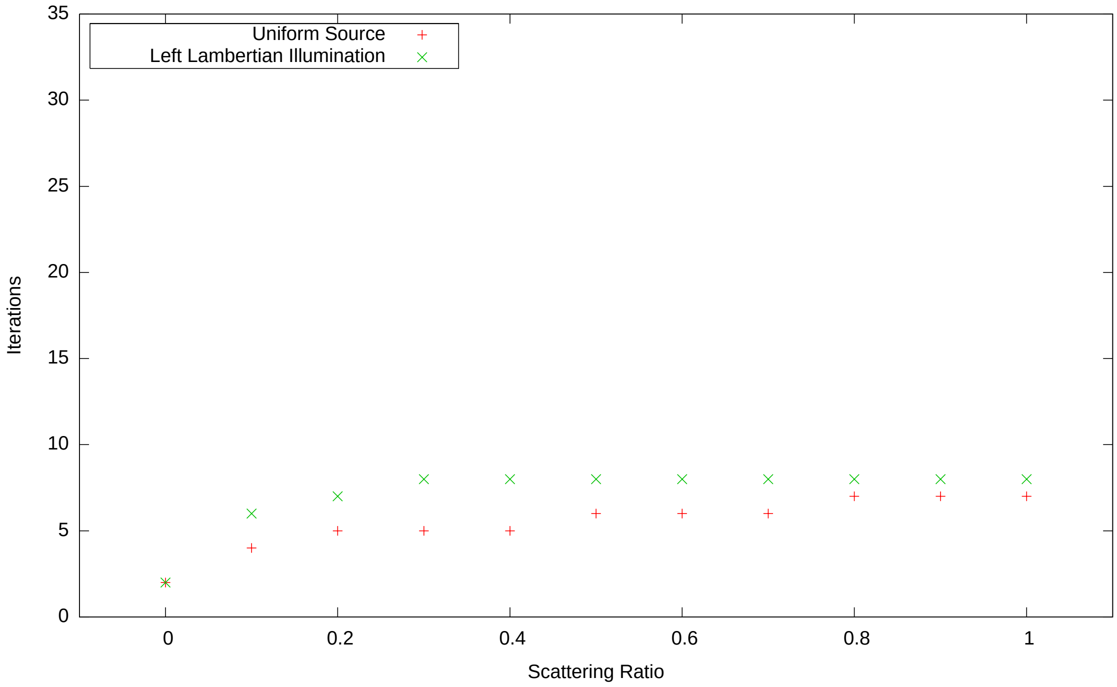


Figure 6.4: Number of Iterations Needed for Convergence Using DI With LD and S_8.

The Lambertian illumination test is more difficult to solve than a uniform source because of coupling of the flux between the left and right halves of the problem and the asymmetry in the flux. When the angular flux is perturbed in the left hand side, it causes a perturbation in the right hand side, which then couples back to the left hand side (provided $c > 0$). The presence of an embedded neutron source masks the effect of perturbation and, hence, allows the problem to converge faster.

6.1.1.5 Tests 8 and 9. For Test 8, both unaccelerated source iteration and TSA required more iterations than DI to converge. DSA had good performance, requiring only 22 to 27 iterations when the problem was smaller than 5 region pairs. When the problem was 5 region pairs and larger, DSA was not providing any benefit and was turned off by PARTISN, thus effectively becoming an unaccelerated method.

DI and PARTISN did not agree to within the convergence tolerance when PARTISN required many iterations to converge. For example, TSA and DI agreed in the one region pair case (123 iterations) and did not agree in the two region pair case

(386 iterations). The source of the discrepancy is difficult to fully ascertain without instrumenting PARTISN. The most likely cause is a combination of false convergence.

For Test 9, DI required fewer iterations to converge (Figure 6.5) than PARTISN. As the problem got larger than 30 region pairs, PARTISN would not perform any additional iterations. The reason for the difference between the solutions produced by DI and PARTISN is due to false convergence on the part of PARTISN. The convergence test that PARTISN performs is

$$\max_{i,g} \left| \frac{\phi_{i,g}^{l+1} - \phi_{i,g}^{l}}{\phi_{i,g}^{l}} \right| < \epsilon, \tag{6.1}$$

where $\phi_{i,g}^{l}$ is the scalar flux at iteration l at mesh point i, group g and ϵ is the convergence tolerance. At 30 region pairs, the scalar flux at the right edge of the problem is 2.08×10^{-24}. Recalling (1.24) and the problem of small contributions as the number of scatters gets large, PARTISN is exhibiting that behavior. With such a small flux, the contribution from neutrons that have over 95 collisions (using unaccelerated source iteration as the measure of scattering contributions) to the scalar flux is smaller than the convergences tolerance.

Overall, agreement between DI and PARTISN for these two tests is acceptable given the limitations of the source iteration method. Further research is needed to verify the solutions generated by DI beyond the regime where PARTISN generates acceptable results.

6.1.1.6 Test 10. Tables 6.2 through 6.5 show the number of iterations required for convergence for test 10 (the Two Region Scattering Ratio Sweep Test). I also ran the test using the LD spatial quadrature and observed no significant difference in the number of iterations. The number of iterations required for convergence was essentially the same for both the DP$_n$ and S$_n$ angular quadratures. In all cases, DI converged in fewer iterations, particularly as $c \rightarrow 1$.

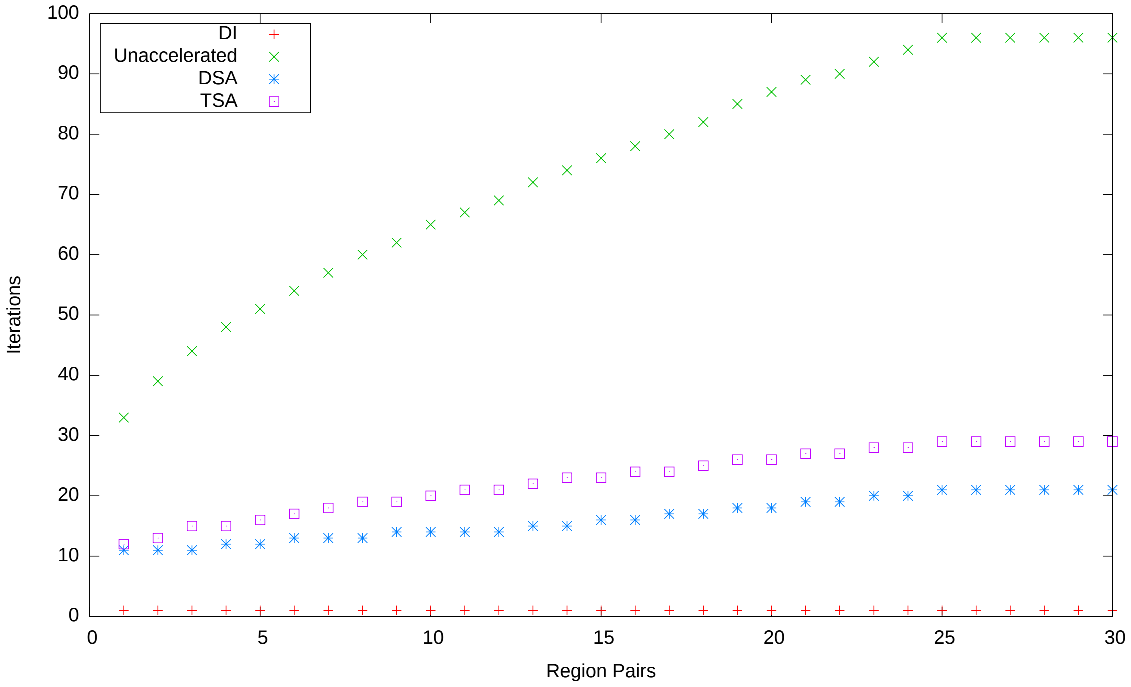


Figure 6.5: Number of Iterations Needed for Convergence for the Periodic Interface, Scattering Ratio Problem Using Diamond-Difference and S_8.

The results shown in tables 6.3 and 6.5 are symmetric relative to the pure absorber to the pure scatter diagonal (a difference of one iteration is ignored). Table 6.2 has a slight asymmetry when the absorbing region is on the left versus the right. When the absorbing region is on the left and the scattering region is on the right, there are two isotropic (or nearly isotropic) flows into the left region. This forces DI to transition from isotropic to an anisotropic flow from two different directions. Conversely, when the absorbing region is on the right and the scattering region is on the left, there is only one isotropic flow into the right region.

The DSA case (Table 6.4) also has an asymmetry relative to the diagonal. When the scattering region is on the left and the absorber is on the right, the problem is similar to the Left Lambertian Illumination test (Test 7). When the right region is a pure absorber, the problem is equivalent to a 2 mfp Left Lambertian Illumination test (which converges in the same number of iterations). As the scattering ratio of the right region increases, the problem evolves into a 4 mfp Left Lambertian Illumination test. When the absorbing region is on the left and the scattering region is on the right, the left region is quickly solved and the right region has an anisotropic illumination on the left.

6.1.2 XY Geometry. A summary of the number of iterations required for convergence is shown in Table 6.6. The performance that DI exhibited when using the diamond-difference spatial quadrature did not reveal any significant performance advantage; in fact, in some cases DI required more iterations. For example, in the case of the pure scatterer test, DI needed more iterations to converge than DSA when using DD; however, with SC and LC, DI required only 9 iterations to converge.

The infinite medium test, as in the slab geometry case, was able to converge within one iteration when using an isotopic distribution as the initial condition. Changing DI to use a random initialization of the inward current distribution prevents convergence when using DD. If SC is used instead of DD, DI is able to converge within 17 iterations.

Table 6.2: Number of iterations for the two region scattering ratio sweep using DI, diamond difference, DP_4 double-range Gauss-Legendre angular quadrature.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	6	6	7	7	7	7	8	8	8	7
	0.1	6	6	6	7	7	7	8	8	8	8	8
	0.2	7	7	7	7	7	7	8	8	8	8	8
	0.3	7	7	7	7	7	7	8	8	8	8	8
	0.4	8	8	8	8	8	8	8	8	8	8	8
	0.5	8	8	8	8	8	8	8	8	8	8	8
	0.6	8	8	8	8	8	8	8	8	8	8	8
	0.7	9	9	9	8	8	8	8	8	8	8	8
	0.8	9	9	9	9	9	9	9	8	8	8	8
	0.9	9	9	9	9	9	9	9	8	8	8	8
	1.0	9	9	9	9	9	9	9	8	8	8	7

Table 6.3: Number of iterations for the two region scattering ratio sweep using SI, no source acceleration, diamond difference, DP_4 double-range Gauss-Legendre angular quadrature.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	8	11	13	16	19	23	28	35	46	63
	0.1	8	9	11	13	16	19	23	28	36	46	64
	0.2	11	11	12	14	16	20	24	29	36	47	65
	0.3	13	13	14	15	17	20	24	29	36	47	66
	0.4	16	16	17	17	19	21	25	30	37	48	67
	0.5	19	19	20	20	21	23	27	32	39	50	70
	0.6	23	23	24	24	25	27	29	34	41	52	73
	0.7	28	28	29	29	30	32	34	38	45	56	78
	0.8	35	35	36	37	38	39	41	45	51	63	87
	0.9	45	46	47	47	49	50	53	56	63	75	103
	1.0	62	63	64	66	68	70	73	78	87	103	141

Table 6.4: Number of iterations for the two region scattering ratio sweep using SI, DSA, diamond difference, DP$_4$ double-range Gauss-Legendre angular quadrature.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	6	7	8	8	9	10	11	12	13	14
	0.1	5	6	7	8	8	9	10	11	12	13	14
	0.2	6	6	7	7	8	9	10	11	12	13	14
	0.3	7	7	7	7	8	9	10	11	12	13	15
	0.4	7	7	7	7	8	9	10	11	12	13	15
	0.5	8	8	8	8	8	9	10	11	12	14	15
	0.6	8	8	8	8	8	9	10	11	12	14	15
	0.7	9	9	9	8	9	9	10	11	12	14	16
	0.8	10	9	9	9	9	9	10	11	13	14	16
	0.9	10	10	10	10	9	9	10	11	13	15	17
	1.0	11	11	11	11	11	10	10	11	13	15	18

Table 6.5: Number of iterations for the two region scattering ratio sweep using SI, TSA, diamond difference, DP$_4$ double-range Gauss-Legendre angular quadrature.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	5	6	7	7	8	8	9	10	13	17
	0.1	5	5	6	7	7	8	8	9	10	13	17
	0.2	6	6	6	7	7	8	8	9	10	13	17
	0.3	7	6	6	7	7	8	8	9	10	13	17
	0.4	7	7	7	7	7	8	8	9	10	13	18
	0.5	8	8	7	7	8	8	8	9	11	13	18
	0.6	8	8	8	8	8	8	9	9	11	14	19
	0.7	9	9	9	9	9	9	9	10	11	14	19
	0.8	10	10	10	10	10	11	11	11	13	15	21
	0.9	13	13	13	13	13	13	14	14	15	18	25
	1.0	17	17	17	17	18	18	19	19	21	25	33

In the two region scattering ratio sweep test, the performance exhibited by DI when using the diamond-difference spatial quadrature (Table 6.7) was significantly worse than PARTISN using DSA (Table 6.11). For some combinations of scattering ratio, DI was unable to converge in fewer than 500 iterations (indicated by "NC" in the table). When compared to unaccelerated source iteration (Table 6.10), DI with diamond difference performs better in most cases.

The eight cases where DI and PARTISN (indicated in bold in Table 6.7) do not have any apparent pattern other than they all occur when the left region has a scattering ratio less than 0.7 and in five of the eight cases the left region had a scattering ratio less than 0.5. When there are no embedded sources in a cell and the scattering ratio is low, negative fluxes are more likely to occur. The negative fluxes will result in a variety of unphysical artifacts, e.g. strong axial flows and oscillations in the angular distribution, which is the most likely cause for the differences in the two solutions. Interestingly enough, only three of the four cases where DI did not converge within 500 iterations are in this set of eight; one of the non-convergent cases was able to produce the same solution as PARTISN. In all eight cases, the area of the problem where the two solutions failed to agree was always along the far right edge. The angular flux in the right half of the problem is much smaller than the angular flux in the left half because the only source of neutrons is from the illumination on the left boundary, thus negative fluxes are more likely to occur.

When the negative flux fix-up scheme was used, the irregular pattern of iteration counts vanished (Table 6.8). There was an increase in iteration count when $c > 0.8$ relative to PARTISN. A direct comparison of the scalar fluxes generated by DI and PARTISN is not possible because the difference in fix-up schemes will cause them to converge to different solutions. Instead, I compared the DI and PARTISN results to the results produced by DI using the step characteristic spatial quadrature. Because step characteristic is non-negative and has second-order convergence, this comparison would highlight the effect of the fix-up on the solution. The magnitude of the variation in the two diamond difference solutions relative to the step characteristic solution was approximately the same. Thus, the solution produced by my fix-up scheme was no worse than the solution produced by PARTISN with its fix-up scheme.

When using a positive spatial quadrature such as step characteristic, the performance of DI improves dramatically (Table 6.9). This behavior further highlights the problem of using a non-positive spatial quadrature with distribution iteration.

6.1.3 XYZ Geometry. The two region sweep problem (Test 10) proved to be very challenging for DI. For virtually all of the combinations of scattering ratios, DI was unable to converge within 50 iterations when using diamond-difference, even after the convergence tolerance was changed to 1.0×10^{-5}. The flux fix-up scheme was able to correct most of the non-convergent cases, however, when $c > 0.7$ the fix-up scheme lost effectiveness in reducing the iteration count.

Table 6.12 summarizes the number of iterations required for convergence. The tests where DI was unable to converge using DD are noted with "NC." When a positive spatial quadrature was used, DI was able to converge.

6.2 Multigroup Verification Tests

Table 6.13 summarizes the number of iterations that each method required to reach the convergence tolerance in slab geometry. DI performed as well, or better, than either unaccelerated source iteration or TSA. DSA converged more rapidly in the "Three Group Absorber, Upscatter" test; however, DSA did not converge to the same solution as DI, unaccelerated source iteration and TSA. The PARTISN log file did not give any indication that it encountered any problems and had to disable DSA. Without performing a detailed analysis of the PARTISN code, it is not clear why DSA did not converge to the same solution.

The results for XY geometry was similar to slab geometry, including the difference in the "Three Group Absorber, Upscatter" test. For the downscatter tests the number of iterations was the same as in the slab geometry case. In XY geometry, my DI code initializes the angular inward current distribution for every outer iteration,

Table 6.6: Number of iterations required for convergence in the time independent, XY geometry tests.

	DI		Source Iteration with DD	
Test Problem	DD	SC	Unaccelerated	DSA
Pure Absorber	2	2	2	2
Pure Scatterer	16	4	586	11
Infinite Medium	1	1	139	106
Constant Source, Single-Region	6	5	18	7
Isotropic Source, 4mfp	31	23	N/A	N/A
Isotropic Source, 32mfp	NC	65	N/A	N/A
Lambertian Source, 4mfp, Vacuum	18	8	22	9

Table 6.7: Number of iterations for the two region scattering ratio sweep using DI, diamond difference, S_4 single-range Gauss-Legendre angular quadrature in XY geometry.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	5	6	8	7	8	9	10	12	11	15
	0.1	8	**51**	14	9	7	9	10	10	12	11	15
	0.2	8	8	9	13	8	14	**NC**	10	11	11	13
	0.3	12	13	8	8	78	11	12	13	12	13	14
	0.4	13	10	12	25	15	**NC**	11	10	11	13	16
	0.5	11	**149**	15	**22**	11	18	10	15	13	13	16
	0.6	16	13	25	19	10	12	20	11	13	14	15
	0.7	**56**	15	17	12	14	12	18	11	24	14	16
	0.8	17	15	14	17	14	12	14	36	14	15	18
	0.9	20	14	18	16	22	**NC**	14	NC	14	17	18
	1.0	19	15	23	**27**	24	17	15	17	17	20	21

Table 6.8: Number of iterations for the two region scattering ratio sweep using DI, diamond difference, S_4 single-range Gauss-Legendre angular quadrature in XY geometry and a negative flux fix-up.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	5	7	6	7	7	9	10	11	11	13
	0.1	5	5	6	7	8	8	9	9	11	11	13
	0.2	6	6	6	7	7	8	8	10	10	11	13
	0.3	6	7	7	7	8	8	9	11	11	11	14
	0.4	8	7	8	8	8	9	10	10	11	12	14
	0.5	8	7	8	8	9	9	10	10	12	12	14
	0.6	9	9	9	9	9	9	13	11	12	13	14
	0.7	10	9	10	10	12	11	11	11	12	14	16
	0.8	10	11	11	11	11	12	13	13	13	15	16
	0.9	12	11	12	13	13	13	13	15	15	15	16
	1.0	13	13	14	14	15	16	16	19	16	19	21

Table 6.9: Number of iterations for the two region scattering ratio sweep using DI, step characteristic, S_4 single-range Gauss-Legendre angular quadrature in XY geometry.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	4	5	5	6	6	7	7	8	8	9
	0.1	5	5	5	6	6	6	7	7	8	9	9
	0.2	5	5	6	6	6	7	7	8	8	9	9
	0.3	6	6	6	6	6	7	7	8	8	9	10
	0.4	6	6	6	7	7	7	8	8	8	9	10
	0.5	7	7	7	7	7	8	8	8	9	9	10
	0.6	7	7	7	8	8	8	8	8	9	10	10
	0.7	8	8	8	8	8	8	9	9	10	10	10
	0.8	9	9	9	9	9	9	9	10	10	10	11
	0.9	9	9	10	10	10	10	10	10	10	11	12
	1.0	10	10	10	10	11	11	11	11	12	12	13

Table 6.10: Number of iterations for the two region scattering ratio sweep using SI, no acceleration, diamond difference, S$_4$ single-range Gauss-Legendre angular quadrature in XY geometry.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	9	11	15	16	19	22	26	32	39	51
	0.1	9	10	12	14	16	19	22	26	32	39	51
	0.2	12	12	13	14	16	19	22	27	32	40	51
	0.3	14	14	14	15	17	20	23	27	33	40	52
	0.4	16	16	16	17	18	20	23	28	33	41	53
	0.5	19	19	19	20	20	22	25	29	34	42	54
	0.6	22	22	22	23	24	25	27	30	35	43	56
	0.7	26	26	27	27	28	29	30	33	38	46	58
	0.8	32	32	32	33	33	34	36	38	42	49	62
	0.9	39	40	40	40	41	42	43	46	49	56	69
	1.0	51	51	52	52	53	54	56	58	62	69	83

Table 6.11: Number of iterations for the two region scattering ratio sweep using SI, DSA, diamond difference, S$_4$ single-range Gauss-Legendre angular quadrature in XY geometry.

		Left Region Scattering Ratio										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Right Region Scattering Ratio	0.0	2	6	8	9	9	10	10	11	11	12	13
	0.1	6	7	8	8	9	9	10	11	12	12	13
	0.2	7	7	7	8	9	9	10	11	12	12	13
	0.3	8	8	8	8	9	9	10	11	12	12	13
	0.4	8	8	8	8	9	9	10	11	12	13	13
	0.5	9	8	8	9	9	9	10	11	12	13	14
	0.6	9	9	9	9	9	9	10	11	12	13	14
	0.7	10	10	9	9	9	9	10	11	12	13	14
	0.8	10	10	10	10	10	10	10	11	12	13	14
	0.9	11	11	11	11	11	11	10	11	12	13	14
	1.0	12	12	12	12	12	12	11	11	12	13	14

Table 6.12: Number of iterations required for convergence in the time independent, XYZ geometry tests.

	DI		Source Iteration	
Test Problem	DD	Step	Unaccelerated	DSA
Pure Absorber	2	2	2	2
Pure Scatterer	37	8	410	11
Infinite Medium	NC	14	421	388
Constant Source, Single-Region	11	8	17	8

thus the total number of iterations in the "Three Group Absorber, Upscatter" test was greater than what PARTISN required (30 for DI and 18 for unaccelerated source iteration).

The multigroup problems were tested using three different initialization strategies for the inward angular current distribution for each energy group: Isotropic at every outer iteration (Isotropic); isotropic for the first outer iteration and the previous distribution for subsequent outer iterations (Previous); and a random initialization for every outer iteration (Random). As shown in Table 6.14, for downscatter problems there is no advantage between Isotropic and the Previous initialization methods. For problems that have upscatter, the Previous method performs better than the Isotropic method. The Random initialization method demonstrates that convergence is not predicated on the initial distribution when solving multigroup problems.

Table 6.13: Total number of inner iterations required for convergence in the time independent, multigroup, slab geometry tests.

		Source Iteration		
Test Problem	DI	Unaccelerated	DSA	TSA
Three Group Absorber, Downscatter	6	6	6	6
Three Group Alternating	11	135	13	34
Three Group Absorber, Upscatter	21	21	9	21

Table 6.14: Total number of inner iterations required for convergence in the time independent, multigroup, slab geometry inward angular partial current initialization method tests.

Test Problem	Isotropic	Previous	Random
Three Group Absorber, Downscatter	6	6	6
Three Group Alternating	11	11	13
Three Group Absorber, Upscatter	24	21	24

6.3 Time Dependent Verification Tests

The agreement between PARTISN and DI was sufficient to demonstrate that DI can be used as the transport code in a time dependent problem. The difference that was observed in the two source ramp tests was due to the differences in the time steps and where the time-varying source was sampled. The number of iterations required for each time step was consistent with the results seen in the time independent testing, e.g. only two iterations are needed when the synthetic scattering ratio was small. For the pure scatter case, when $v\Delta t$ was increased, thereby allowing $\tilde{c} \rightarrow c$, the number of iterations increased. Also, when the synthetic cross section was large, the number of iterations was generally insensitive to changes in the actual total cross section because the synthetic scattering ratio was relatively unaffected. Both DI and PARTISN exhibited these behaviors, which supports the conclusion that DI performs in an equivalent fashion as PARTISN.

VII. Conclusion

The Distribution Iteration method has consistently been demonstrated as a reliable replacement for source iteration. Prior to my research, there had been no formal verification plan for distribution iteration. The previous work by Wager and Prins only implemented tests designed to aid in the development of the algorithm. The verification plan that was implemented did not reveal any unexpected difference between the results produced by DI and accelerated source iteration.

7.1 Contributions

In my research, I did demonstrate that DI can be used to solve the time dependent transport equation using the same method used by production source iteration codes. As demonstrated in this dissertation, there is no optimal combination of parameters, e.g. time step and cell size, for producing reliable results. The choice of a time step becomes more complicated with multigroup problems. While it is possible to pick a time step and still obtain a reasonable cell optical thickness for monoenergetic problems, a multigroup problem requires a transport algorithm that tolerates optically thick cells for some energy groups. My research demonstrated that DI was able to produce reasonable results for optically thick time dependent problems without a fix-up mechanism.

One of the design goals for the software that I developed as part of my research was to be able to support new research. I implemented a flexible and modular code, which also incorporated real time visualization. Given the uniqueness of the DI algorithm, I had to develop the visualization methods.

In order to be able to use the TIEL benchmark developed by Ganapol, Mathews and I developed the matrix albedo for time independent problems. This new method provides an accurate solution for the effect that an infinite medium will have on the angular flux in a region of interest. While some aspects of the implementation were straightforward, I implemented all boundary conditions, e.g. specular reflection, in matrix form. This new approach provides a flexible method for handling difficult boundary conditions, such as asymmetric angular quadratures.

Several improvements to the DI algorithm are original to my research. The first improvement was the methodical implementation of the three geometries. This implementation is more compact and maintainable than previous versions and was instrumental in keeping verification plan simple. The improvement to the angular current distribution refinement algorithm converges more rapidly than the previous algorithm. The new algorithm is also more parallel efficient than the previous version. Finally, the partial current solver can easily be adapted to different linear algebra solvers and has improved parallel efficiency.

7.2 Conclusions

Wager and Prins performed initial assessments of the relative performance between DI and SI. My research examined a broader range of the parameter space using both DSA and TSA. Based on my research, DI consistently performed as well or better than either DSA or TSA in slab geometry. In XY and XYZ geometries, DI continues to exhibit superior performance when using positive spatial quadratures. The performance of DI relative to DSA and TSA supports the conclusion that DI is a powerful replacement for source iteration. For the classes of problems where DSA loses effectiveness and TSA fails, DI exhibits robust performance and converges in fewer iterations.

In XYZ geometry, DI does demonstrate a pronounced weakness when using a spatial quadrature that generates negative fluxes. The poor performance exhibited with diamond difference is strong evidence that DI should be used with a positive spatial quadrature such as SC or with a spatial quadrature that is negative less frequently than DD, e.g. LC or LD. I did implement a new negative angular partial current distribution fix-up, which is different then the fix-up that PARTISN performs, however, it was ineffective in highly scattering problems ($c > 0.7$).

For time dependent transport, PARTISN implements an adaptive time step algorithm and a negative flux fixup. In addition, PARTISN enables the negative flux fixup algorithm in the transport equation. Reverse engineering these three algorithms was impractical and of little value. The comparisons were adequate to confirm that my time-dependent DI code has no errors worse than PARTISN's.

The poor performance that DI exhibited when using DD is not a failure of the DI algorithm. Mathews [26] demonstrated the unphysical ray effect that DD exhibits, which calls into question the reliability of the results generated by DD when negative fluxes occur, independent of the algorithm. A key advantage that DI has is that new spatial quadratures can be implemented easily while implementing a new spatial quadrature in DSA is not trivial.

Currently, there is no acceleration implemented in the DI algorithm. In slab and XY geometry there was little motivation to add any acceleration when using DI because it outperforms DSA and TSA. In XYZ geometry my research shows that acceleration will have little benefit when using a positive spatial quadrature. There may be some classes of problems where DI may benefit from a convergence accelerator.

Appendix A. Other Transport Codes

A.1 TIMEX

The within-group BTE used by the 1972 TIMEX code [30] is originally presented as (N.B. some subscripts have been eliminated for clarity)

$$\frac{1}{v}\left(\frac{\psi_i^{j+1}-\psi_i^{j}}{\Delta t}\right)+\mu_m\left(\frac{A_{i+1/2}\psi_{i+1/2}^{j+1}-A_{i-1/2}\psi_{i-1/2}^{j+1}}{V_i}\right)$$
$$+\left(\frac{\alpha_{m+1/2}\psi_{m+1/2}^{j+1}-\alpha_{m-1/2}\psi_{m-1/2}^{j+1}}{w_m V_i}\right)+\sigma_i\psi_i^{j+1}=S_{m,i}^{j}+q_{m,i}. \quad \text{(A.1)}$$

TIMEX 1972 uses a cell-centered indexing scheme, where j is the time index, i is the cell index and m is the ordinate index. Cell faces, therefore, are indicated by a half step, e.g. $i+1/2$ is the cell face between cells i and $i+1$. The area of a cell face is represented by A and the cell volume by V. The ordinates and weights are represented by μ and w, respectively. The variable α represents the curvature coefficient. The angular flux is represented by ψ, S is the scattering and fission sources from the j time interval, and q is the inhomogeneous source. Finally, v is the group velocity and Δt is the time step.

The combination of A, V and α allows the TIMEX 1972 code to solve three different one-dimensional geometries–slab, cylindrical, and spherical. The curvature coefficient is defined recursively as

$$\alpha_{m+1/2}-\alpha_{m-1/2}=\mu_m w_m\left(A_{i+1/2}-A_{i-1/2}\right), \quad \text{(A.2)}$$

using $\alpha_{1/2}=\alpha_{N+1/2}=0$, where N is the number of ordinates, as starting conditions (N.B. the ordinate index, m starts at 1). The definitions for A and V are shown in table A.1. Note that a recursive definition is used for the area elements in spherical geometry. This was done to improve accuracy for cells near the center of the sphere. Also, in cylindrical geometry the ordinates represented by μ_m and w_m is actually a two-dimensional angular quadrature of μ and ξ.

Table A.1: Definition of geometric functions used by TIMEX 1972.

Geometry	Variable	Cell Face Area (A)	Cell Volume (V)
Slab	$x_{i+1/2}$	1	$x_{i+1/2} - x_{i-1/2}$
Cylindrical	$r_{i+1/2}$	$2\pi r_{i+1/2}$	$\pi\left(r^2_{i+1/2} - r^2_{i-1/2}\right)$
Spherical	$r_{i+1/2}$	$\frac{2V_i}{r_{i+1/2}-r_{i-1/2}} - A_{i-1/2}$	$\frac{4\pi}{3}\left(r^3_{i+1/2} - r^3_{i-1/2}\right)$

TIMEX (1972) uses a cell average relationship between the cell-centered flux and the edges in both space and angle

$$\psi^{j+1}_{i,m} = \frac{\psi^{j+1}_{i+1/2} + \psi^{j+1}_{i-1/2}}{2} \tag{A.3}$$

$$\psi^{j+1}_{i,m} = \frac{\psi^{j+1}_{m+1/2} + \psi^{j+1}_{m-1/2}}{2}. \tag{A.4}$$

These relationships are used eliminate the $i + 1/2$ and $m + 1/2$ for $\mu_m > 0$ directions and $i - 1/2$ and $m - 1/2$ for $\mu_m < 0$ directions and obtain

$$\begin{aligned}\left(\frac{V_i}{v\Delta t} + 2\mu_m A_{i+1/2} + \frac{2\alpha_{m+1/2}}{w_m} + \sigma_i V_i\right)\psi^{j+1}_i &= \mu_m\left(A_{i+1/2} + A_{i-1/2}\right)\psi^{j+1}_{i-1/2}\\ &+ \left(\frac{\alpha_{m+1/2} + \alpha_{m-1/2}}{w_m}\right)\psi^{j+1}_{m-1/2} + \left(\frac{V_i}{v\Delta t}\right)\psi^j_i + S^j_{m,i}V_i + q_{m,i}V_i.\end{aligned} \tag{A.5}$$

The recursion relationship for the curvature coefficient is used to obtain

$$\begin{aligned}&\left(\frac{V_i}{v\Delta t} + \mu_m\left(A_{i+1/2} + A_{i-1/2}\right) + \frac{\alpha_{m+1/2} + \alpha_{m-1/2}}{w_m} + \sigma_i V_i\right)\psi^{j+1}_i\\ &\quad= \mu_m\left(A_{i+1/2} + A_{i-1/2}\right)\psi^{j+1}_{i-1/2} + \left(\frac{\alpha_{m+1/2} + \alpha_{m-1/2}}{w_m}\right)\psi^{j+1}_{m-1/2}\\ &\qquad+ \left(\frac{V_i}{v\Delta t}\right)\psi^j_i + S^j_{m,i}V_i + q_{m,i}V_i.\end{aligned} \tag{A.6}$$

Because my research is focused on Cartesian geometry, the BTE simplifies to (substituting $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and dividing by Δx_i)

$$\left(\frac{1}{v\Delta t} + 2\frac{\mu_m}{\Delta x_i} + \sigma_i\right)\psi_i^{j+1} = 2\frac{\mu_m}{\Delta x_i}\psi_{i-1/2}^{j+1} + \frac{1}{v\Delta t}\psi_i^j + S_{m,i}^j + q_{m,i}. \tag{A.7}$$

Substituting the relationship for the midpoint spatial average and algebraically rearranging the terms yields

$$\left(\frac{1}{v\Delta t} + \sigma_i\right)\psi_i^{j+1} + \mu_m\frac{\psi_{i+1/2}^{j+1} - \psi_{i-1/2}^{j+1}}{\Delta x_i} = \frac{1}{v\Delta t}\psi_i^j + S_{m,i}^j + q_{m,i}. \tag{A.8}$$

The use of the diamond-difference spatial discretization can result in a negative outgoing angular flux. Using the midpoint spatial average to express ψ_i^{j+1} in terms of $\psi_{i+1/2}^{j+1}$ and $\psi_{i-1/2}^{j+1}$ yields

$$\left(\frac{1}{v\Delta t} + \sigma_i\right)\frac{\psi_{i+1/2}^{j+1} + \psi_{i-1/2}^{j+1}}{2} + \mu_m\frac{\psi_{i+1/2}^{j+1} - \psi_{i-1/2}^{j+1}}{\Delta x_i} = \frac{1}{v\Delta t}\psi_i^j + S_{m,i}^j + q_{m,i}. \tag{A.9}$$

Solving for $\psi_{i+1/2}^{j+1}$ in terms of $\psi_{i-1/2}^{j+1}$ yields the equation

$$\psi_{i+1/2}^{j+1} = \frac{1 - \frac{\Delta x}{2\mu}\left(\sigma_i + \frac{1}{v\Delta t}\right)}{1 + \frac{\Delta x}{2\mu}\left(\sigma_i + \frac{1}{v\Delta t}\right)}\psi_{i-1/2}^{j+1} + \frac{\frac{1}{v\Delta t}\psi_i^j + S_{m,i}^j + q_{m,i}}{1 + \frac{\Delta x}{2\mu}\left(\sigma_i + \frac{1}{v\Delta t}\right)}. \tag{A.10}$$

The $\psi_{i-1/2}^{j+1}$ coefficient will be negative when

$$\sigma_i\Delta x + \frac{\Delta x}{v\Delta t} > 2|\mu|. \tag{A.11}$$

When $\sigma_i + \frac{1}{v\Delta t} >> 1$, the source term will be negligible and either $\sigma_i\Delta x$ or $\frac{\Delta x}{v\Delta t}$ (or both) will be large. Consider the case where $\Delta t \to 0$,

$$\psi_{i+1/2}^{j+1} = \frac{2\mu}{\Delta x}\psi_i^j - \psi_{i-1/2}^{j+1}. \tag{A.12}$$

Thus, $\psi^{j+1}_{i+1/2}$ has the potential for being negative if the cell centered flux is small, particularly when μ is small.

Removing the discretization of the spatial derivative from equation A.8 yields

$$\left(\frac{1}{v\Delta t} + \sigma_i + \mu_m \frac{\partial}{\partial x}\right) \psi_i^{j+1} = \frac{1}{v\Delta t}\psi_i^j + S^j_{m,i} + q_{m,i}. \tag{A.13}$$

Similarly, the discretization of the time derivative can be removed, which yields

$$\left(\frac{1}{v}\frac{\partial}{\partial t} + \sigma_i + \mu_m \frac{\partial}{\partial x}\right) \psi_i^{j+1} = S^j_{m,i} + q_{m,i}. \tag{A.14}$$

In operator notation, this is equivalent to

$$\mathbb{V}^{-1}\frac{\partial}{\partial t}\Psi^{j+1} + L\Psi^{j+1} = S\Psi^j + q, \tag{A.15}$$

where

$$L = \mu_m \frac{\partial}{\partial x} + \sigma_i. \tag{A.16}$$

A.2 PARTISN

PARTISN is based on the implicit in time form of the BTE. Starting from the equation (Alcouffe [4])

$$\left(\frac{1}{v}\frac{\partial}{\partial t} + \hat{\Omega}\cdot\nabla + \sigma\right)\psi = S \tag{A.17}$$

and using a central difference for the time derivative yields

$$\frac{1}{v\Delta t}\left(\psi^{j+1/2} - \psi^{j-1/2}\right) + \hat{\Omega}\cdot\nabla\psi^j + \sigma\psi^j = S^j. \tag{A.18}$$

The above equation, which is equivalent to the explicit midpoint method, has second order local truncation error in time, i.e. $O((\Delta t)^2)$. The equation can be returned to an implicit form by eliminating $\psi^{j+1/2}$. Consider the midpoint time average

$$\psi^j = \frac{\psi^{j+1/2} + \psi^{j-1/2}}{2}. \tag{A.19}$$

The midpoint time average can be rewritten to solve $\psi^{j+1/2}$ as

$$\psi^{j+1/2} = 2\psi^j - \psi^{j-1/2}. \tag{A.20}$$

We can eliminate $\psi^{j+1/2}$ by using the midpoint time average obtaining,

$$\frac{2}{v\Delta t}\left(\psi^j - \psi^{j-1/2}\right) + \hat{\Omega} \cdot \nabla\psi^j + \sigma\psi^j = S^j. \tag{A.21}$$

The above equation is now a backwards difference and has first-order local truncation error. Regrouping the terms in a similar fashion to the TIMEX 1976 approach yields

$$\hat{\Omega} \cdot \nabla\psi^j + \left(\sigma + \frac{2}{v\Delta t}\right)\psi^j = S^j + \frac{2}{v\Delta t}\psi^{j-1/2}. \tag{A.22}$$

The flux at the $j+1/2$ time step is recovered by using equation (A.20) combined with the result for ψ^j obtained from equation (A.22).

The time stepping algorithm implemented by PARTISN is shown in figure A.1. Because PARTISN was adapted from the DANTSYS transport code, using half time steps for the extrapolated flux was notationally convenient. The half time steps are computed by the midpoint extrapolation and the integer intervals are computed by the transport algorithm. The notation is, however, awkward and can be expressed instead as

$$\hat{\Omega} \cdot \nabla\psi^j + \left(\sigma + \frac{1}{v\Delta t'}\right)\psi^j = S^j + \frac{1}{v\Delta t'}\psi^{j-1}, \tag{A.23}$$

where $\Delta t' = \Delta t/2$. The extrapolation step then becomes

$$\psi^{j+1} = 2\psi^j - \psi^{j-1}. \tag{A.24}$$

It is important to note that the truncation error associated with the ψ^j angular flux is first order because it is backward Euler–there is no apparent benefit in the truncation error to using this approach. The improvement in the truncation error is in the extrapolation step. Doing a Taylor series expansion on ψ^{j+1} and ψ^{j-1} yields

$$\begin{aligned}\psi^{j+1} &= \psi^j + \Delta t' \frac{\partial \psi}{\partial t}\Big|_j + \frac{(\Delta t')^2}{2}\frac{\partial^2 \psi}{\partial t^2}\Big|_j + O((\Delta t')^3) && \text{(A.25)}\\ \psi^{j-1} &= \psi^j - \Delta t' \frac{\partial \psi}{\partial t}\Big|_j + \frac{(\Delta t')^2}{2}\frac{\partial^2 \psi}{\partial t^2}\Big|_j - O((\Delta t')^3). && \text{(A.26)}\end{aligned}$$

Substituting in the Taylor series expansion for ψ^{j-1} into the extrapolation step yields

$$\psi^{j+1} = \psi^j + \Delta t' \frac{\partial \psi}{\partial t}\Big|_j - \frac{(\Delta t')^2}{2}\frac{\partial^2 \psi}{\partial t^2}\Big|_j + O((\Delta t')^3). \tag{A.27}$$

Note that the extrapolation step results in a ψ^{j+1} that is accurate to $(\Delta t')^2$ term. Thus, this method will yield second-order local truncation error only with the extrapolated fluxes and not with the midpoint fluxes.

PARTISN, however, will only output fluxes (scalar or angular) from the midpoint time and not the extrapolated end of time step flux. Thus, the results from PARTISN are only accurate to first order even though the algorithm is accurate to second order internally. PARTISN solves (A.22), which is effectively Backwards Euler, in the transport solver and then, in a separate state, applies the extrapolation step. One advantage to this approach is that the global error will scale by $(\Delta t)^2$ rather than by Δt if backward Euler was used.

```
Initialize $\psi^{1/2}$ to the starting angular flux
for j = 1 to Number of Time Steps do
  Compute $\psi^j$ from the transport equation
  Use the midpoint average to extrapolate to $\psi^{j+1/2}$
  Increment the time index
  The extrapolated flux becomes the new $\psi^{j-1/2}$ angular flux
end for
```

Algorithm A.1: Semi-Implicit Time Stepping Algorithm

In order to conduct the testing to support my research, two changes to the PARTISN code were made. The first change was to have PARTISN preserve the scalar flux (RTFLUX) for each time step. By default, PARTISN discards the scalar flux from each time step and there is no parameter file option to change that behavior. The second change was to correct an erroneous time stamp in the time dependent scalar flux. PARTISN was writing the end of time step rather than the midpoint time, which was the scalar flux that was being written to the file.

Appendix B. Transport Coefficients

Wager [32] and Prins [28] derived the transport coefficients in slab and XY geometries for various spatial quadratures. In the following sections, I will present the derivation of the Step method spatial quadrature in XY and XYZ geometries and Diamond-Difference in XYZ geometry.

The derivations are shown using the time independent form of the BTE. If time dependent coefficients are needed for the synthetic time form of the BTE, both the cross section term, σ and the average source term, S_A, are replaced with the corresponding synthetic form.

B.1 Step Method

The Step method is defined by the assumption that the angular flux is constant within a cell (supressing the cell index number for clarity), specifically

$$\psi_{\mathrm{R}} = \psi^{\mathrm{A}} \quad \mu > 0 \tag{B.1}$$

$$\psi_{\mathrm{L}} = \psi^{\mathrm{A}} \quad \mu < 0 \tag{B.2}$$

and

$$\psi_{\mathrm{T}} = \psi^{\mathrm{A}} \quad \eta > 0 \tag{B.3}$$

$$\psi_{\mathrm{B}} = \psi^{\mathrm{A}} \quad \eta < 0, \tag{B.4}$$

where μ and η are the direction cosines from the angular quadrature. This spatial quadrature is very inaccurate, but it does have the benefit of being positive and useful when debugging. The derivation begins with the zeroth-moment balance equation

$$\mu \frac{\psi_R - \psi_L}{\Delta x} + \eta \frac{\psi_T - \psi_B}{\Delta y} + \sigma \psi_A = S_A. \tag{B.5}$$

Applying the constant flux assumption to eliminate ψ_R and ψ_T and grouping like terms yields

$$\left(\frac{\mu}{\Delta x} + \frac{\eta}{\Delta y} + \sigma\right)\psi_A = S_A + \frac{\mu}{\Delta x}\psi_L + \frac{\eta}{\Delta y}\psi_B. \tag{B.6}$$

Let

$$\epsilon_x = \frac{\sigma \Delta x}{\mu}, \tag{B.7}$$

$$\epsilon_y = \frac{\sigma \Delta y}{\eta} \quad \text{and} \tag{B.8}$$

$$\alpha = \frac{\epsilon_y}{\epsilon_x}. \tag{B.9}$$

Note that α is effective aspect ratio for the cell for that ordinate. Multiplying Equation B.6 by $\Delta y/\eta$ yields

$$\left(\epsilon_y + \alpha + 1\right)\psi_A = S_A\frac{\Delta y}{\eta} + \alpha\psi_L + \psi_B. \tag{B.10}$$

Solving for ψ_A yields

$$\psi_A = \frac{S_A\frac{\Delta y}{\eta} + \alpha\psi_L + \psi_B}{\epsilon_y + \alpha + 1}. \tag{B.11}$$

The transport coefficients for the contributions to the cell average angular flux are

$$\mathbb{K}_x^{\psi\mathrm{I}} = \frac{\alpha}{\epsilon_y + \alpha + 1}, \tag{B.12}$$

$$\mathbb{K}_y^{\psi\mathrm{I}} = \frac{1}{\epsilon_y + \alpha + 1}, \tag{B.13}$$

$$\mathbb{K}^{\psi\mathrm{S}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + 1} \quad \text{and} \tag{B.14}$$

$$\mathbb{K}^{\psi\mathrm{E}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + 1}. \tag{B.15}$$

The coefficients for the contributions to the X faces (left and right faces) are

$$\mathbb{K}_{xx}^{\mathrm{OI}} = \frac{\alpha}{\epsilon_y + \alpha + 1}, \tag{B.16}$$

$$\mathbb{K}_{xy}^{\mathrm{OI}} = \frac{1}{\epsilon_y + \alpha + 1}, \tag{B.17}$$

$$\mathbb{K}_{x}^{\mathrm{OE}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + 1} \quad \text{and} \tag{B.18}$$

$$\mathbb{K}_{x}^{\mathrm{OE}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + 1}. \tag{B.19}$$

The coefficients for the contributions to the Y faces (bottom and top faces) are

$$\mathbb{K}_{yx}^{\mathrm{OI}} = \frac{\alpha}{\epsilon_y + \alpha + 1}, \tag{B.20}$$

$$\mathbb{K}_{yy}^{\mathrm{OI}} = \frac{1}{\epsilon_y + \alpha + 1}, \tag{B.21}$$

$$\mathbb{K}_{y}^{\mathrm{OS}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + 1} \quad \text{and} \tag{B.22}$$

$$\mathbb{K}_{y}^{\mathrm{OE}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + 1}. \tag{B.23}$$

The derivation for the X-Y-Z geometry is similar, with the solution for the cell average angular flux as

$$\psi_A = \frac{S_A \frac{\Delta y}{\eta} + \alpha\psi_L + \psi_B + \beta\psi_P}{\epsilon_y + \alpha + \beta + 1}, \tag{B.24}$$

where $\beta = \epsilon_y/\epsilon_z$ and $\epsilon_z = \sigma\Delta z/\xi$. The transport coefficients for the contributions to the cell average angular flux are

$$\mathbb{K}_x^{\psi\mathrm{I}} = \frac{\alpha}{\epsilon_y + \alpha + \beta + 1}, \tag{B.25}$$

$$\mathbb{K}_y^{\psi\mathrm{I}} = \frac{1}{\epsilon_y + \alpha + \beta + 1}, \tag{B.26}$$

$$\mathbb{K}_z^{\psi\mathrm{I}} = \frac{\beta}{\epsilon_y + \alpha + \beta + 1}, \tag{B.27}$$

$$\mathbb{K}^{\psi\mathrm{S}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1} \quad \text{and} \tag{B.28}$$

$$\mathbb{K}^{\psi\mathrm{E}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1}. \tag{B.29}$$

The coefficients for the contributions to the X faces (left and right faces) are

$$\mathbb{K}_{xx}^{\mathrm{OI}} = \frac{\alpha}{\epsilon_y + \alpha + \beta + 1}, \tag{B.30}$$

$$\mathbb{K}_{xy}^{\mathrm{OI}} = \frac{1}{\epsilon_y + \alpha + \beta + 1}, \tag{B.31}$$

$$\mathbb{K}_{xz}^{\mathrm{OI}} = \frac{\beta}{\epsilon_y + \alpha + \beta + 1}, \tag{B.32}$$

$$\mathbb{K}_x^{\mathrm{OS}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1} \quad \text{and} \tag{B.33}$$

$$\mathbb{K}_x^{\mathrm{OE}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1}. \tag{B.34}$$

The coefficients for the contributions to the Y faces (bottom and top faces) are

$$\mathbb{K}_{yx}^{\mathrm{OI}} = \frac{\alpha}{\epsilon_y + \alpha + \beta + 1}, \tag{B.35}$$

$$\mathbb{K}_{yy}^{\mathrm{OI}} = \frac{1}{\epsilon_y + \alpha + \beta + 1}, \tag{B.36}$$

$$\mathbb{K}_{yz}^{\mathrm{OI}} = \frac{\beta}{\epsilon_y + \alpha + \beta + 1}, \tag{B.37}$$

$$\mathbb{K}_{y}^{\mathrm{OS}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1} \quad \text{and} \tag{B.38}$$

$$\mathbb{K}_{y}^{\mathrm{OE}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1}. \tag{B.39}$$

The coefficients for the contributions to the Z faces (back and front faces) are

$$\mathbb{K}_{zx}^{\mathrm{OI}} = \frac{\alpha}{\epsilon_y + \alpha + \beta + 1}, \tag{B.40}$$

$$\mathbb{K}_{zy}^{\mathrm{OI}} = \frac{1}{\epsilon_y + \alpha + \beta + 1}, \tag{B.41}$$

$$\mathbb{K}_{zz}^{\mathrm{OI}} = \frac{\beta}{\epsilon_y + \alpha + \beta + 1}, \tag{B.42}$$

$$\mathbb{K}_{z}^{\mathrm{OS}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1} \quad \text{and} \tag{B.43}$$

$$\mathbb{K}_{z}^{\mathrm{OE}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + \alpha + \beta + 1}. \tag{B.44}$$

B.2 Diamond Difference

The Diamond Difference spatial quadrature applies the assumption that the angular fluxes are related to the cell average angular flux by

$$\psi_L + \psi_R = \psi_B + \psi_T = \psi_F + \psi_P = 2\psi_A \tag{B.45}$$

as an auxiliary equation to the cell balance equation

$$\mu\frac{\psi_R - \psi_L}{\Delta x} + \eta\frac{\psi_T - \psi_B}{\Delta y} + \xi\frac{\psi_F - \psi_P}{\Delta z} + \sigma\psi_A = S_A. \tag{B.46}$$

Using Equation B.45 to solve for ψ_R, ψ_T, and ψ_F in terms of ψ_L, ψ_B, ψ_P and ψ_A yields

$$\psi_R = 2\psi_A - \psi_L, \tag{B.47}$$

$$\psi_T = 2\psi_A - \psi_B \quad \text{and} \tag{B.48}$$

$$\psi_F = 2\psi_A - \psi_P. \tag{B.49}$$

Using the above equations to eliminate ψ_R, ψ_T and ψ_F and grouping like terms yields

$$\left(\sigma + 2\frac{\mu}{\Delta x} + 2\frac{\eta}{\Delta y} + 2\frac{\xi}{\Delta z}\right)\psi_A = S_A + 2\frac{\mu}{\Delta x}\psi_L + 2\frac{\eta}{\Delta y}\psi_B + 2\frac{\xi}{\Delta z}\psi_P. \tag{B.50}$$

Let

$$\epsilon_x = \frac{\sigma \Delta x}{\mu}, \tag{B.51}$$

$$\epsilon_y = \frac{\sigma \Delta y}{\eta} \tag{B.52}$$

$$\epsilon_z = \frac{\sigma \Delta z}{\xi} \tag{B.53}$$

$$\alpha = \frac{\epsilon_y}{\epsilon_x} \quad \text{and} \tag{B.54}$$

$$\beta = \frac{\epsilon_y}{\epsilon_z}. \tag{B.55}$$

The ratios α and β was chosen to be consistent with the X-Y diamond difference equation derived by Mathews [26]. Multipling Equation B.50 by $\Delta y/\eta$ yields

$$(\epsilon_y + 2\alpha + 2 + 2\beta)\,\psi_A = S_A\frac{\Delta y}{\eta} + 2\alpha\psi_L + 2\psi_B + 2\beta\psi_P. \tag{B.56}$$

Solving for the cell average scalar flux provides

$$\psi_A = \frac{S_A\frac{\Delta y}{\eta} + 2\alpha\psi_L + 2\psi_B + 2\beta\psi_P}{\epsilon_y + 2\alpha + 2 + 2\beta}. \tag{B.57}$$

Solving for ψ_R, ψ_T and ψ_F yields

$$\psi_R = \frac{2S_A\frac{\Delta y}{\eta} + (2\alpha - \epsilon_y - 2 - 2\beta)\psi_L + 4\psi_B + 4\beta\psi_P}{\epsilon_y + 2\alpha + 2 + 2\beta} \tag{B.58}$$

$$\psi_T = \frac{2S_A\frac{\Delta y}{\eta} + 4\alpha\psi_L + (2 - \epsilon_y - 2\alpha - 2\beta)\psi_B + 4\beta\psi_P}{\epsilon_y + 2\alpha + 2 + 2\beta} \tag{B.59}$$

$$\psi_F = \frac{2S_A\frac{\Delta y}{\eta} + 4\alpha\psi_L + 4\psi_B + (2\beta - \epsilon_y - 2\alpha - 2)\psi_P}{\epsilon_y + 2\alpha + 2 + 2\beta} \tag{B.60}$$

Using the above solutions, the transport coefficients for the contributions to the cell average angular flux are

$$\mathbb{K}_x^{\psi\mathrm{I}} = \frac{2\alpha}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.61}$$

$$\mathbb{K}_y^{\psi\mathrm{I}} = \frac{2}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.62}$$

$$\mathbb{K}_z^{\psi\mathrm{I}} = \frac{2\beta}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.63}$$

$$\mathbb{K}^{\psi\mathrm{S}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \quad \text{and} \tag{B.64}$$

$$\mathbb{K}^{\psi\mathrm{E}} = \frac{\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.65}$$

$$\tag{B.66}$$

The coefficients for the contributions to the X faces (left and right faces) are

$$\mathbb{K}_{xx}^{\mathrm{OI}} = \frac{2\alpha - \epsilon_y - 2 - 2\beta}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.67}$$

$$\mathbb{K}_{xy}^{\mathrm{OI}} = \frac{4}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.68}$$

$$\mathbb{K}_{xz}^{\mathrm{OI}} = \frac{4\beta}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.69}$$

$$\mathbb{K}_{x}^{\mathrm{OS}} = \frac{2\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \quad \text{and} \tag{B.70}$$

$$\mathbb{K}_{x}^{\mathrm{OE}} = \frac{2\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.71}$$

$$\tag{B.72}$$

The coefficients for the contributions to the Y faces (bottom and top faces) are

$$\mathbb{K}_{yx}^{\mathrm{OI}} = \frac{4\alpha}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.73}$$

$$\mathbb{K}_{yy}^{\mathrm{OI}} = \frac{2 - \epsilon_y - 2\alpha - 2\beta}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.74}$$

$$\mathbb{K}_{yz}^{\mathrm{OI}} = \frac{4\beta}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.75}$$

$$\mathbb{K}_{y}^{\mathrm{OS}} = \frac{2\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \quad \text{and} \tag{B.76}$$

$$\mathbb{K}_{y}^{\mathrm{OE}} = \frac{2\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.77}$$

$$\tag{B.78}$$

The coefficients for the contributions to the Z faces (back and front faces) are

$$\mathbb{K}_{zx}^{\mathrm{OI}} = \frac{4\alpha}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.79}$$

$$\mathbb{K}_{zy}^{\mathrm{OI}} = \frac{4}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.80}$$

$$\mathbb{K}_{zz}^{\mathrm{OI}} = \frac{2\beta - \epsilon_y - 2\alpha - 2}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.81}$$

$$\mathbb{K}_{z}^{\mathrm{OS}} = \frac{2\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \quad \text{and} \tag{B.82}$$

$$\mathbb{K}_{z}^{\mathrm{OE}} = \frac{2\frac{\Delta y}{\eta}}{\epsilon_y + 2\alpha + 2\beta + 2} \tag{B.83}$$

$$\tag{B.84}$$

Currents that are incident on the boundary of the region of interest can be classified by the treatment of the source term. Consider the following simplified, linearized BTE for a pure absorber ($c = 0$) medium,

$$\mu \frac{\partial \psi}{\partial x} + \sigma \psi = Q(x, \mu). \tag{C.1}$$

For illustrative purposes, we can further simplify this example by assuming that the incident current is from a source on the left boundary and there is no source of neutrons internal to the problem, thus all the neutrons are moving in the $\mu > 0$ direction. A current source on the left can be expressed as

$$Q(x, \mu) = \begin{cases} Q(\mu) & \text{when } x = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{C.2}$$

We can group the current sources into two categories based on whether or not there is any μ dependence in the source.

A Lambertian or diffuse source [6] is one that has the same angular flux when viewed from any angle, e.g. the detectors at A and B as shown in Figure C.1 both observe the same angular flux from the differential surface element dA. Thus, a Lambertian source has a uniform flux per steradian–$Q(\mu)$ = constant. Solving equation C.1) for positive μ yields

$$\psi(x, \mu) = Q e^{-x\sigma/\mu}. \tag{C.3}$$

Integrating over all μ yields the scalar flux, ϕ,

$$\phi(x) = \frac{Q}{2} \left(e^{-x\sigma} - x\sigma \Gamma(0, x\sigma) \right), \tag{C.4}$$

where $\Gamma(a, z)$ is the upper incomplete Gamma function [1] defined as

$$\Gamma(a, z) = \int_x^\infty t^{a-1} e^{-t} \mathrm{d}t\ . \tag{C.5}$$

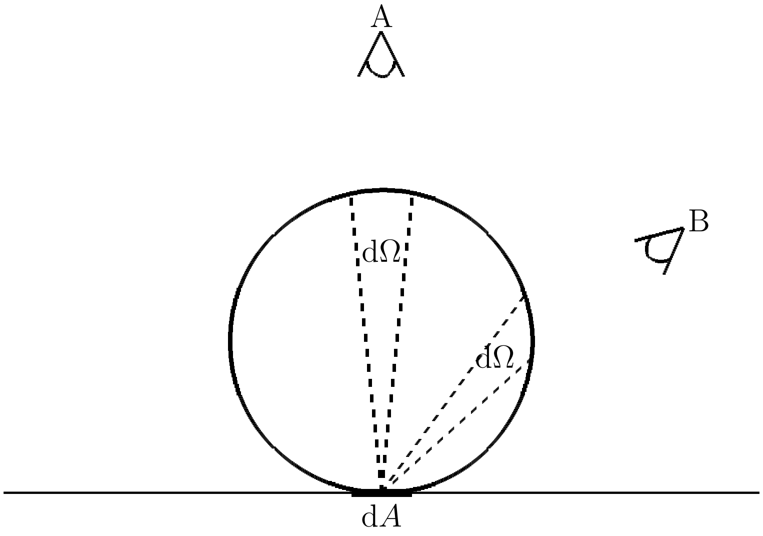


Figure C.1: Lambertian Source

As the left boundary is approached, $\phi(x) \rightarrow Q/2$. In terms of current, j, along an ordinate a Lambertian source will have the familiar cosine dependence: $j = \mu Q$. Integrating over $\mu > 0$, the rightward (positive) partial current on the left face is $Q/4$ (using the normalization scheme introduced by Lewis and Miller [22]).

There is one specific type of source of the second type that is useful in benchmarking transport codes–those that produce a uniform current source, viz. $j =$ constant. This source is characterized by the uniform current flowing from the surface, hence the appellation *isotropic surface source* (ISS). The angular flux for this type of source is

$$\psi(x, \mu) = \frac{Q}{\mu} e^{-x\sigma/\mu} \tag{C.6}$$

with the scalar flux (for positive x) being

$$\phi(x) = \frac{Q}{2} \Gamma(0, x\sigma). \tag{C.7}$$

As the left boundary is approached, $\phi(x) \rightarrow \infty$. The rightward partial current on the left face is $Q/2$.

The key differences to note between the Lambertian and the isotropic surface sources are the partial currents ($Q/4$ versus $Q/2$, respectively) and the finite versus infinite scalar flux at the current surface, as shown in Figure C.2. Based on these two distinctions, one can quickly verify how a particular method has treated an incident source. For example, PARTISN utilizes a Lambertian current source.

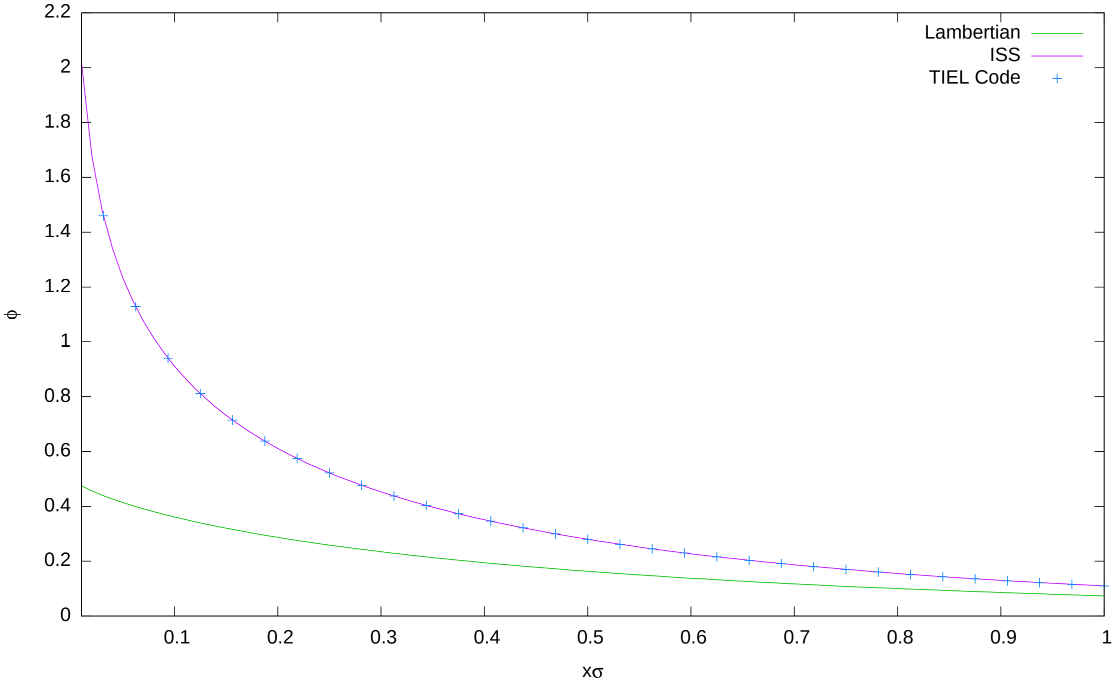


Figure C.2: Lambertian Illumination and Isotropic Surface Source Scalar Fluxes

In the case of Ganapol's TIEL benchmark, the angular flux in the pure absorber case is

$$\psi(x,\mu) = \frac{e^{-x\sigma/\mu}H(x\sigma)}{2\mu}, \tag{C.8}$$

where $H(z)$ is the Heaviside step function. Integration over μ yields the scalar flux

$$\phi(x) = \frac{1}{2}\Gamma(0,x\sigma)H(x\sigma). \tag{C.9}$$

We can conclude that the TIEL benchmark will behave identically to the isotropic surface source near the origin. Figure C.2 also shows the plot of the scalar fluxes from the TIEL benchmark.

Bibliography

1. Abramowitz M. and Stegun I. A., editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* (Ninth Edition). New York, NY: Dover, 1970.

2. Alcouffe R. E., Baker R. S., Dahl J. A., Turner S. A., and Ward R. C. *PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System*. Technical Report LA-UR-05-3925, Los Alamos, NM: Los Alamos National Laboratory, May 2005.

3. Alcouffe R. E. "Diffusion Synthetic Acceleration Methods For the Diamond-Differenced Discrete Ordinates Equations," *Nuclear Science and Engineering*, *64*:344 (1977).

4. Alcouffe R. E. and Baker R. S. *Time-Dependent Deterministic Transport on Parallel Architectures Using PARTISN*. Technical Report LA-UR-98-272, Los Alamos, NM: Los Alamos National Laboratory, July 1998.

5. Azmy Y. Y. "Unconditionally Stable and Robust Adjacent-Cell Diffusive Preconditioning of Weighted-Difference Particle Transport Methods Is Impossible," *Journal of Computational Physics*, *182*:213–233 (2002).

6. Born M. and Wolf E. *Principles of Optics* (7th Edition). Cambridge: Cambridge University Press, 2002.

7. Carlson B. G. *Solution of the Transport Equation by S_n Approximations*. Technical Report LA-1599, Los Alamos, NM: Los Alamos Scientific Laboratory, 1953.

8. Carlson B. G. and Lathrop K. D. *Computing Methods in Reactor Physics*. Gordon and Breach, 1968.

9. Carlson B. G. and Lathrop K. D. *The Method of Discrete Ordinates*. Technical Report LA-3251-MS, Los Alamos, NM: Los Alamos Scientific Laboratory, October 1965.

10. Castrianni C. L. and Adams M. L. "A Nonlinear Corner-Balance Spatial Discretization for Transport on Arbitrary Grids." *Proceedings International Conference on Mathematics and Computations, Reactor Physics and Environmental Analyses2*. 916. 1995.

11. Dupree S. A., Sandmeier H. A., Hansen G. E., Engle Jr. W. W., and Mynatt F. R. *Time-Dependent Neutron and Photon Transport Calculations Using the Method of Discrete Ordinates*. Technical Report LA-4557, Los Alamos, NM: Los Alamos Scientific Laboratory, May 1971.

12. Ganapol B. D., Baker R. S., Dahl J. A., and Alcouffe R. E. *Homogeneous Infinite Media Time-Dependent Analytical Benchmarks*. Technical Report LA-UR-01-1854, Los Alamos, NM: Los Alamos National Laboratory, 2001.

13. Ganapol B. D. *Analytical Radiation Transport Benchmarks for The Next Century*. Technical Report DOE/ID/14113, University of Arizona, 2005.

14. Goluoglu S. and Dodds H. L. "A Time-Dependent, Three-Dimensional Neutron Transport Methodology," *Nuclear Science and Engineering*, *139*:248–261 (2001).

15. Hill T. R. *ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of the One-Dimensional Multigroup Transport Equation*. Technical Report LA-5990-MS, Los Alamos, NM: Los Alamos National Laboratory, 1975.

16. Hill T. R. and Reed W. H. *TIMEX: A Time-Dependent Explicit Discrete Ordinates Program for the Solution of Multigroup Transport Equations with Delayed Neutrons*. Technical Report LA-6201-MS, Los Alamos, NM: Los Alamos Scientific Laboratory, February 1976.

17. Iserles A. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, 1996.

18. Larsen E. W. and Alcouffe R. E. *Linear characteristic method for spatially discretizing the discrete ordinates equations in (X,Y)-geometry*. Technical Report LA-UR-81-101, Los Alamos, NM 87545: Los Alamos Scientific Laboratory, 1981.

19. Larsen E. W. and Nelson P. "Finite-Difference Approximations and Superconvergence For the Discrete-Ordinate Equations In Slab Geometry," *SIAM Journal on Numerical Analysis*, *19*(2):334–348 (April 1982).

20. Lathrop K. D. "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy," *Journal of Computational Physics*, *4*:475–498 (1969).

21. Lathrop K. D., Anderson R. E., and Brinkley F. W. *TRANZIT: A Program for Multigroup Time-Dependent Transport in* (ρ, z) *Cylindrical Geometry*. Technical Report LA-4575, Los Alamos, NM: Los Alamos Scientific Laboratory, February 1971.

22. Lewis E. E. and Miller Jr. W. F. *Computational Methods of Neutron Transport*. American Nuclear Society, Inc., La Grange Park, IL, 1993.

23. Mathews K. and Dishaw J. "Matrix Albedo For Discrete Ordinates Infinite-Medium Boundary Condition." *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*. 2007.

24. Mathews K., Prins N., Wager N., and Dishaw J. "Distribution Iteration–An Alternative to Source Iteration for Deterministic Radiation Transport Methods," *Nuclear Science and Engineering*, *Submitted Aug 2007* (2007).

25. Mathews K., Sjoden G., and Minor B. "Exponential Characteristic Spatial Quadrature for Discrete Ordinates Radiation Transport in Slab Geometry," *Nuclear Science and Engineering*, *118*:24 (1994).

26. Mathews K. A. "On the Propagation of Rays in Discrete Ordinates," *Nuclear Science and Engineering*, *132*:155–180 (1999).

27. Miller D. J. *Linear characteristic spatial quadrature for discrete ordinates neutral particle transport on arbitrary triangles*. PhD dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH 45434, 1993.

28. Prins N. J. *Distribution Iteration: A Robust Alternative to Source Iteration for Solving the Discrete Ordinates Radiation Transport Equations In Slab and XY-Geometries*. PhD dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, 2006.

29. Reed W. H. "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," *Nuclear Science and Engineering*, *45*:245 (1971).

30. Reed W. H. *TIMEX: A Time-Dependent Explicit Discrete Ordinates Program for the Solution of Multigroup Transport Equations*. Technical Report LA-4800, Los Alamos, NM: Los Alamos Scientific Laboratory, July 1972.

31. Thacker B. H., Doebling S. W., Hemez F. M., Anderson M. C., Pepin J. E., and Rodriguez E. A. *Concepts of Model Verification and Validation*. Technical Report LA-14167-MS, Los Alamos, NM 87117: Los Alamos National Laboratory, October 2004.

32. Wager N. J. *A Rapidly-Converging Alternative to Source Iteration For Solving the Discrete Ordinates Radiation Transport Equations In Slab Geometry*. PhD dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 2004.

33. Walters W. F. and Wareing T. A. "An Accurate, Strictly-Positive, Nonlinear Characteristic Scheme for the Discrete-Ordinates Equations," *Transport Theory and Statistical Physics*, *25*:197–215 (1996).

34. Wareing T. A. *An Exponential Discontinuos Scheme for Discrete-Ordinate Calculations in Cartesian Geometries*. Technical Report LA-UR-97-2269, Los Alamos, NM 87545: Los Alamos National Laboratory, 1997.

Glossary

error A defect due to an incomplete implementation of the model and is not due to lack of knowledge., 61

explicit method A numerical method for solving differential equations where only current time step values are used to compute extrapolated values, 14

fidelity The degree to which a model is an accurate representation of physical process., 60

fully discretized A numerical method for solving partial differential equations where all derivatives are represented with finite differences, 14

implicit method A numerical method for solving differential equations where values from multiple time steps, including extrapolated values, are used to compute extrapolated values, 15

isotropic surface source A current source where the current along each ordinate is constant. See Lambertian source., 120

Lambertian source A current source which has uniform radiance per steradian, 119

positive method A spatial quadrature that will unconditionally generate positive fluxes, 79

semi-discretization A numerical method for solving partial differential equations where only the spatial derivatives are represented with finite differences, 14

semi-implicit An implicit method that has been solved by linearization, 16

stiff equation A differential equation that is numerically unstable when the step size is not sufficiently small, 15

uncertainty A potential defect in the model due to insufficient fidelity to the physical process., 61

validation A process for assessing the suitability of a model in its intended application., 60

verification A process for assessing the conformance of a model's implementation., 60

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 2007-10-03	**2. REPORT TYPE** PhD Dissertation	**3. DATES COVERED** *(From — To)* Sept 2004 — Sep 2007

4. TITLE AND SUBTITLE Time Dependent Discrete Ordinates Neutron Transport Using Distribution Iteration in XYZ Geometry	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) James Dishaw, Maj	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/DS/ENP/07-S01
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Technical Applications Center 1030 S. Highway A1A Patrick AFB FL 32925-3002	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The DI algorithm is an alternative to source iteration that, in our testing, does not require an accelerator. I developed a formal verification plan and executed it to verify the results produced by my code that implemented DI with the above features. A new, matrix albedo, boundary condition treatment was developed and implemented so that infinite-medium benchmarks could be included in the verification test suite. The DI algorithm was modified for parallel efficiency and the prior instability of the refinement sweep was corrected. The testing revealed that DI performed as well or faster than source iteration with DSA and that DI continued to work where DSA failed. Performance did degrade when the diamond-difference (without fixup) spatial quadrature was used. Because diamond-difference is a non-positive spatial quadrature, it can produce nonphysical negative fluxes, particularly in higher dimensions. I developed a new fixup scheme to accommodate the negative fluxes, but it did not improve performance in XYZ geometry when the scattering ratio was near unity.

15. SUBJECT TERMS

Radiation Transport, Boltzmann Transport Equation, Discrete Ordinates, Deterministic Transport, Time Dependent, Multigroup

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr Kirk Mathews
U	U	U	UU	146	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255–3636, ext 4508